THE OPTIMAL LOCATION OF GEODSS SENSORS IN CANADA

THESIS

Pierre J. Forgues, Major, CF

AFIT/GSO/ENS/91M-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(1)

AFIT/GSO/ENS/91M-1

THE OPTIMAL LOCATION OF
GEODSS SENSORS IN CANADA

THESIS

Pierre J. Forgues, Major, CF

AFIT/GSO/ENS/91M-1

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
February 1991

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
The Optimal Location of GEODSS Sensors in Canada

5. FUNDING NUMBERS

6. AUTHOR(S)
Pierre J. Forgues, Major, Canadian Forces

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSO/ENS/91M-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Defense Headquarters
Directorate of Air Requirements (DAR 6)
Ottawa, Canada K1A-0K2

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The research presents a study of the p-median and set covering facility location problems as applied to the task of finding the optimal location(s) of Ground Based Electro-Optical Deep Space Surveillance System (GEODSS) sensors in Canada to maximize coverage of satellites of interest. A second criterion function, consisting of minimizing the sum of the variances in coverage from selected locations, is then added to the formulations. Deficiencies of MOLP (multi-objective linear programming) techniques in generating the efficient frontier of an integer problem are revealed. A FORTRAN 77 solution algorithm is developed to identify non-dominated alternatives, and generate an ordered list of feasible alternatives using paired comparisons with the ideal.

A case study is presented which shows that a minimum of two southerly locations are generally required to adequately cover the geostationary belt above Canada. The study reveals that while two-site solutions therefore exclude northerly locations, three-site solutions will usually include northerly locations which optimize coverage of highly inclined synchronous orbits.

14. SUBJECT TERMS
Space Surveillance, Mathematical Programming, GEODSS
Multicriteria Optimization, Facility Location, P-Median

15. NUMBER OF PAGES
304

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFI1/GSO/ENS/9'M-1

THE OPTIMAL LOCATION OF GEODSS SENSORS IN CANADA

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Space Operations

Pierre J. Forgues, B.S., M.A.
Major, Canadian Forces

February 1991

THESIS APPROVAL

STUDENT: Maj Pierre J. Forgues CLASS: GSO 91M

THESIS TITLE: The Optimal Location of GEODSS Sensors in Canada

DEFENSE DATE: 22 February 1991

COMMITTEE:

Advisor: Dr Yupo Chan/ENS Yupo Chan

Reader: Dr T.S. Kelso/ENS Thomas Sean Kelso

ACKNOWLEDGEMENTS

Working on a thesis without an advisor, would be like attempting to land an aircraft blindfolded. I was especially lucky to have as mentor an accomplished tutor and gifted intellectual. Thanks, Dr Yupo Chan, for making this experience enjoyable, rewarding and formative.

Several people have assisted in this effort over the past year, from the development of the research question, to the publication of the final copy. Foremost among these are those who provided the data for the research. These include my reader, Dr T.S. Kelso, and the meteorologists at ETAC/DNY, Captains Anthony J. Warren, James T. Kroll, and John A. Rupp.

Finally, to my wife Karen, whose support never wavered, and to Michelle and Robbie, who now have a full time Dad again, thanks for your love and understanding.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

The research presents a study of the maximal coverage p-median and of the set covering facility location problems as applied to the GEODSS location problem. The classical single-objective mathematical formulations of the p-median and set covering problems are converted into network-flow formulations and various solution methodologies are developed using a scaled-down version of the GEODSS problem.

The next step of the research is the introduction of a second criterion function into the problem. This second function consists of minimizing the sum of the variance in coverage at the selected locations. The research reveals the deficiencies of MOLP (multiobjective linear programming) techniques in generating the efficient frontier of an integer problem. A "brute-force" solution algorithm is developed and programmed in FORTRAN 77 to generate all feasible alternatives, determine which of these are non-dominated, and then provide an ordered list of alternatives using paired comparisons with the ideal.

A case study is presented which shows the difficulty in finding a feasible one-site solution given the need to observe a wide segment of the geostationary belt. The example also shows that, for a given satellite population, the optimal alternative must for similar reasons include two southerly locations. The example reveals that while two-site solutions therefore exclude northerly locations, three-site solutions will usually include a northerly location.

THE OPTIMAL LOCATION OF GEODSS SENSORS IN CANADA

I. INTRODUCTION

1. Background

The military mission of space surveillance, as defined in United States Space Command (USSPACECOM) Regulation 55-12, is to detect, track, identify, and catalog all man-made objects in space (46:13). To accomplish this mission, a network of 26 ground-based sensors dispersed around the globe provide observational data to the Space Surveillance Center (SSC) located in the Cheyenne Mountain Complex in Colorado Springs. The SSC computers analyze this data to determine and catalog the location of all man-made satellites in space.

Canada's participation in the space surveillance mission dates back to 1961 with the installation of the first Canadian Baker-Nunn system (an optical film camera) in Cold Lake, Alberta. Later, in 1976, a second Canadian Baker-Nunn site was established in St Margarets, New Brunswick. Space surveillance operations have since been discontinued in Cold Lake but continue today in St Margarets (13:139).

The Ground-Based-Electro-Optical Deep-Space Surveillance (GEODSS) system (an optical video camera) is the replacement for the Baker-Nunn. Four GEODSS systems are located around the world today with a fifth system (GEODSS 5) scheduled for installation in Portugal. At the request of the Commander in Chief of

USSPACECOM, the Canadian Forces have undertaken to operate the St Margarets Baker-Nunn until GEODSS 5 is operational (42).

Canadian participation in the space surveillance mission once GEODSS 5 is operational is being debated by USSPACECOM and Canadian National Defence Headquarters (NDHQ). Within the conte. ıf these discussions, the option of an autonomous network of GEODSS sensors located in Canada is being considered. The NDHQ Directorate of Air Requirements (DAR), the sponsor for this research, has requested that the best locations in Canada for a one-site, two-site, and three-site GEODSS network be determined (18).

2. Research Problem

The objective of this research is to develop a methodology to determine the optimal sensor location(s) of a one-site, two-site, and three-site GEODSS sensor network in Canada.

2.1 Research Sub-Objectives. To attain the above research objective, the following sub-objectives needed to be accomplished:

a. Establish a list of candidate site locations.

b. Obtain the probability that the weather is favorable to GEODSS observations at each of the candidate locations for every month of the year.

   - Obtain GEODSS system specifications.
   - Obtain climatological data.

c. Determine the number of observation opportunities at each candidate location for every month of the year, for every satellite type.

   - Define an observation opportunity.
   - Define the satellite target population.
   - Obtain software to determine the transit frequency.

d. Calculate the expected number of useable observation opportunities at each candidate location for every month of the year based on the probability of favorable weather.

e. Formulate multiple-criteria facility-location and models to determine the optimal location of the GEODSS sensor(s).

   - Define the alternative space.
   - Define the criteria set.
   - Generate the non-dominated set.
   - Define an appropriate preference structure.

3. Scope

The emphasis of this research is to provide the sponsor with a "user-friendly" solution algorithm. The methodology is presented in Chapter VI through the use of an example calculation using actual data. The end-product is a FORTRAN-based software package that orders the feasible set by ascending order of the Manhattan metric distance to the ideal. The ideal is defined as the best attainable value of two defined criterion functions. For each alternative, the Manhattan metric deviation is the sum of the differences from the ideal of each criterion function.

4. Assumptions and Limitations

An assumption is made that operations research scientists at NDHQ will be available to assist in the execution and interpretation of the solution algorithm. Also, coordination with Meteorological Officers is crucial to the selection of candidate locations and the determination of climatological probabilities.

The research does not address political factors that might arise in selecting optimal locations for the tracking of satellites.

II. LITERATURE REVIEW

The purpose of this chapter is to present a synopsis of recent scientific literature that is pertinent to this research. Included are discussions on multiple criteria optimization, facility location theory, network-flow programming, climatology, optical transmission theory, and orbital mechanics. These discussions are not intended as comprehensive treatment of the subject areas, but rather are introductions to some of the terminology and concepts that will appear in later chapters of this report.

1. Multiple Criteria Optimization

1.1 The General Case. The general form of a multiple objective programming problem, given k objective functions, $f_k(x)$, can be stated as follows (2:1):

$$\begin{aligned} &Max\ f_1(x) = z_1 \\ &Max\ f_2(x) = z_2 \\ &\quad \cdots \\ &Max\ f_k(x) = z_k \\ &s.t. \quad x \in S \end{aligned}$$

A trivial solution to the above problem is the vector x which is contained in the feasible region S and which simultaneously maximizes all k objectives. Except for this trivial case, each solution vector in the feasible region will satisfy each of the objectives at varying levels (2:3). Generally speaking, the optimal solution to the problem will represent the solution which offers the permutation that is most appealing to the decision maker.

A solution technique to address the non-trivial case can begin with an assessment of the decision maker's utility function (2:3). This utility function essentially provides an assessment of the value the decision maker assigns to a given solution vector. The multiple objective problem is thus reduced to the single objective problem of maximizing the utility function, U (2:3):

$$\begin{gathered} Max\ \{\ U(z_1, z_2, \ldots, z_k)\ \} \\ s.t.\ f_i(x) = z_i \qquad 1 \le i \le k \\ x \in S \end{gathered}$$

1.2 Definitions and Terminology. The following definitions are taken from Chan (9) and Yu (50):

1.2.1 Multicriteria Decision Making (MCDM). MCDM involves four elements:

(1) Alternative set X, also referred to as the X-space.

(2) Criteria set f_q (if there is only one criterion the problem reduces to a traditional math programming problem)

(3) Outcome set Y, also referred to as the Y-space.

(4) Preference structure, with which the decision maker picks the best outcome.

1.2.2 Versions of MCDM. Multiattribute Decision Analysis (MADA), is the descriptive version of MCDM. Multicriteria Optimization (MCO) is the prescriptive version of MCDM.

1.2.3 Attributes. Attributes are measurable objectives or sub-objectives.

1.2.4 Goal. A goal is a specified level of an attribute; some goals are self-suggested and "more is better" while others are standards to be achieved. We call the former *goal seeking* and the latter *goal setting.*

1.2.5 Criteria. Criteria refer collectively to the attributes, objectives, and goals relative to a specific decision maker in specific situations.

1.2.6 Value/Utility Function. A value/utility function approach to MCDM is to, (a) capture the total value/utility function of the decision maker for the range of possible outcomes associated with alternatives under consideration, and (b) select the alternative(s) that maximizes the decison maker's expected value/utility function.

1.2.7 Satisficing. Satisficing is the process of eliminating alternatives with unacceptable attribute values; while *dominance* is the process of eliminating dominated alternatives. An alternative is said to be dominated when there exists another alternative in the outcome space that is preferred. Conversely, an alternative is non-dominated if its preferred set is empty.

1.2.8 Compromise Solution. Compromise solution is an alternative closest to the "perceived" ideal solution, y^*. The solution methodology proposed at Chapter III of this paper defines the ideal solution from a goal seeking (more is better) perspective. The ideal thus assumes the best feasible value of each criterion function as coordinates in the Y-space. The distance to the ideal is computed using a Manhattan metric.

1.2.9 Pareto Preference. For each criterion function, let greater values be more preferred (i.e., more is better), and assume that no other information on the preference is available or established. Then with respect to Pareto preference, alternative y^1 is preferred to alternative y^2 iff component wise $y^1{}_i \geq y^2{}_i$, i=1,...,q, where "q" is the number of criterion functions.

1.2.10 Ordering. The simplest case of MCDM is simple ordering among alternatives where no preference structure is required. Examples include dominance and Pareto preference. An outcome "y" is pareto optimal iff it is a nondominated solution with respect to Pareto preference. A Pareto optimal solution is also called an efficient, non-inferior, nondominated, or admissible solution.

1.3 Interactive Procedures. Interactive programming methods are useful in solving problems where the decision maker's utility function cannot be completely defined or expressed (28:197).

1.3.1 Description. Interactive techniques involve the decision maker in the solution process. The process is initialized by presenting a limited set of feasible solutions to the decision maker for consideration. The act of choosing a preferred alternative from the limited set provides additional insight into the decision maker's preference structure. Based on this new insight, a new set of solutions is generated and again presented for evaluation. This iterative process is repeated until a stopping criteria is met (28:198; 46:1214).

1.3.2 Classifications. Vanderpooten classifies interactive procedures into two distinct types: search-oriented and learning-oriented procedures (46:1218).

Search-oriented procedures assume that the decision maker's preference structure "pre-exists and remains stable" (46:1218). However, as stated above, an interactive approach is required because the preference structure is not defined and is internal to the decision maker. At each iteration of the process, the decision maker is asked to supply an assessment of the value that should be placed on the current proposal and also to suggest a way of improving this proposal. The process is terminated using a "classical convergence test" (46:1218).

Learning-oriented procedures differ from search-oriented procedures in that assumptions are not made about the stability or even the prior existence of the decision maker's preference structure (46:1218). In fact, through an interactive process similar to the search-oriented procedure, the decision maker plays an important role in the development and formulation of the preference structure. However, contrary to the search-oriented procedure, the decision maker "is free to change his mind and to conduct his exploration in a trial and error fashion" (46:1218). Furthermore, and also unlike the search-oriented procedure, mathematical convergence cannot be achieved and the stopping rule is invoked when the decision maker is satisfied with the exploration of the feasible set (46:1219).

The research problem in this paper appears to satisfy the assumptions of an interactive search-oriented solution methodology. Interviews with experienced orbital analysts, who could qualify as decision makers for this problem, have revealed consistent views on the value of individual objectives (5; 26). However, while a stable utility function appears to exist, this function cannot be readily expressible in mathematical terms.

1.4 Examples of Interactive Applications. There is much written in the literature about interactive multiple- objective programming. The journals surveyed offer innovative techniques for initializing the process and for directing the search for the optimal solution. Three representative methodologies are included here.

1.4.1 Example One. Ringuest and Rinks present two search-oriented interactive procedures for solving multiobjective transportation problems. A transportation problem is a classical linear programming problem where a product must be transported from each of m sources to any of n destinations such that one or more objectives are optimized (36:96).

The first algorithm begins by optimizing each of the objective functions separately but subject to the multi-objective problem constraints to maintain feasibility. This produces an initial set of nondominated solutions which span the solution space (36:100). A nondominated solution represents a point in the feasible space where it is not possible to increase the value of one of the objectives without decreasing the value of another (39:4). An "optimal linear compromise solution" is also obtained and provides more complete coverage of the solution space (36:98). As the name implies, this additional solution is a feasible compromise to the ideal solution (where all objectives are maximized) which is infeasible except for the trivial case.

The decision maker is presented with the above set of solutions and asked to choose a preferred solution. If the decision maker is satisfied with the chosen solution the algorithm is terminated. Otherwise, nondominated solution points, adjacent to the best current solution, are generated and presented for review. The process is repeated until the decision maker is satisfied (36:100).

The authors suggest a modified multicriteria simplex method, which was developed by Yu and Zeleny, to produce the new set adjacent points required at each iteration:

> The Yu and Zeleny algorithm can be used to enumerate all nondominated solutions for a general multiobjective linear program. Multicriteria simplex proceeds from an initial nondominated solution by solving a nondominance subproblem for each adjacent basis. ... A modification by Klingman and Mote ... reduces the computational effort involved in implementing ... the method. (36:98)

The second technique offered by Ringuest and Rinks also begins by optimizing each objective separately to generate a set of nondominated solutions. The decision maker is then asked to review the set of nondominated solutions and the process stops immediately if the decision maker judges one of the solutions to be satisfactory. Otherwise, a function which passes through each of the current nondominated solution vectors is identified. This function has the following form (36:100):

$$z'(x) = \sum_{k=1}^{l} w_k \, z_k(x)$$

The w_k in the above equation represent the weights associated with each nondominated solution vector z_k. These weights are determined by solving the following L by $(L + 1)$ homogeneous system of equations (36:100):

$$\sum_{j=1}^{l} w_k z_{kj} - w_{l+1} = 0, \quad k = 1,\ldots,L$$

where z_{kj} is the jth element of z_k

The function z'(x) can now be optimized using any efficient, single-objective transportation problem algorithm (36:100). If the optimal solution to this problem is preferred to at least one of the current set of nondominated solutions, the new solution is substituted for the least preferred solution and the entire process is repeated. Otherwise, the decision maker chooses the most preferred solution from the current set and the process terminates.

The primary difference between the above two techniques is the method used to generate a new set of nondominated solutions for decision maker appraisal. The first technique searches "along the edges of the feasible decision variable space" while the second algorithm, which relies on a weighting scheme of the objective functions, moves "back and forth across the objective function space" (36:104).

The authors underline the fact that the first algorithm can potentially produce extremely large sets of nondominated solutions (i.e., all adjacent solution points) for review by the decision maker. However, the second algorithm produces exactly (L + 1) alternatives at each iteration, where L is the number of objectives. Therefore, "unless a problem has a large number of objectives, the second algorithm imposes less of a burden on the decision maker" (36:1013).

1.4.2 Example Two. Michalowski presents a learning-oriented technique for solving multiobjective problems.

The starting point of the Michalowski technique is an estimate of the worst solution to the problem (28:198). This differs markedly from the Ringuest and Rinks methodologies presented above which are initialized using solution vectors that include the best values that can be attained for each objective function. Michalowski suggests that using the worst outcome as a reference point is ideally suited to decision situations where "the search for an admissible decision is driven by the desire to avoid undesirable consequences" (28:198).

Michalowski offers three ways of obtaining the initial worst outcome. The simplest technique is to have the decision maker specify the worst outcome levels. Another method is to determine the worst feasible value of each individual objective function. Alternatively, Michalowski suggests that estimates of the worst levels be extracted from a payoff table (28:199).

The general form of a payoff table is shown at Table 1. The entries along a row represent the value, z_{ij}, obtained for objective function z_j when objective function z_i is optimized. Therefore, the entries along the main diagonal form the vector of maximum values for each objectives. The vector formed by taking the minimum value in each column represents an estimate of the worst feasible solution (39:267).

Steuer warns that the payoff table is not a reliable method for obtaining the true minimum values. Computational experience has shown that, in a majority of cases, one or more of the minimum values obtained in this manner are incorrect (39:268). Therefore, the impact of this phenomenon on the Michalowski algorithm should be reviewed before the payoff table method for generating the initial solution is adopted.

Once the initial worst outcome is defined, the iterative process can begin. At each stage of the process, the decision maker compares the set of solutions with the worst case and generates a new decision. The decision is used as a basis to define a new worst case solution which displaces the previous worst outcome. Thus, the decisions taken at each iteration provide information about a decision maker's preferences. This preference information is modelled mathematically to generate

	z_1	z_2		z_k
z^1	z_1^*	z_{12}		z_{1k}
z^2	z_{21}	z_2^*	.	z_{2k}
			.	
z^k	z_{k1}	z_{k2}	.	z_k^*

(39:267)

Table 1 - Payoff Table

another alternative which better meets the decision maker's preferences. The search process can be terminated either when the decision maker is satisfied with the solution or when the generated solutions start to repeat themselves (28:199).

Michalowski claims that one of the main advantages of this technique over other interactive approaches is that "the complexity of the interaction with a decision maker is kept at a minimum" (28:202). Another strength of the approach is that, since the search is learning-oriented, the decision maker is permitted to freely sample the solution space while learning about his/her preference structure (28:202).

1.4.3 Example Three. Arbel and Oren present a search-oriented algorithm to solve multiobjective linear programming problems.

The technique uses the simplex method to generate an initial feasible solution, and to produce the adjacent nondominated solutions for comparison. The search direction for follow-on iterations is generated using a technique called the Analytic Hierarchy Process (2:370).

At each step of the process, the relative weights of the adjacent solution vectors, w_i, and the weight of the current solution vector, w_o, are obtained through pairwise comparison performed by the decision maker. If $w_o \geq w_i$ for all $i = \{1,...,k\}$, then the current solution is the one most preferred by the decision maker and the process terminates. Otherwise the following system of equations (in matrix form) is solved to obtain the gradient vector, *del* $W(y_o)^T$ (2:370):

$$\begin{vmatrix} w_1 - w_o \\ w_2 - w_o \\ \\ w_k - w_o \end{vmatrix} = \begin{vmatrix} y_1^T - y_o^T \\ y_2^T - y_o^T \\ \\ y_k^T - y_o^T \end{vmatrix} \; del\, W(y_o)^T$$

where y_k are the candidate solution vectors
and w_k are the relative weights of these vectors

The original objective functions are weighted using the components of the gradient vector (2:371). The simplex technique is then used to solve the resulting single-objective problem and generate a new current solution to begin the process anew. As indicated above, the stopping criteria is met when the decision maker considers that the current solution carries more weight than all adjacent solutions.

Arbel and Oren claim that a major advantage of this technique is that the decision maker does not have to provide answers to "implicit preference questions concerning his objectives, but instead considers explicit evaluation of adjacent possible improvements" (2:373).

1.5 Evaluation of Interactive Techniques. The authors surveyed generally agree about the usefulness of interactive techniques in solving multiobjective problems.

Steuer claims that "the future of multiple-objective programming is in its interactive application" (39:361). Vanderpooten supports Steuer's assertion and also cites Kok as proposing that "it is nowadays accepted that the interactive approach is the most appropriate way of obtaining the preferences of a decision maker" (46:1217). Steuer also provides the following insight:

> Interactive procedures permit an effective division of labor. They allow the computer to do what it does best (process data and execute algorithms), and they allow the decision maker to do what he or she does best (make improved judgments in the face of new information). (39:361)

Interactive techniques are facilitated by the use of computers. Gibson et al. advances that "specifically, the use of computer graphics may greatly facilitate the process of interactive decision making" (15:104). Also, the ability to examine multiple scenarios and replay a number of "what-if" scenarios serve to enhance the solution process (15:104).

Gibson et al. also explain that solution techniques may be problem-specific and point to the need to wisely select the appropriate multiple objective algorithm (15:104). In a more recent article, Mote and Venkataramanan suggest a set of criteria that should be used for evaluating interactive solution techniques. First, the technique chosen should enhance the decision maker's understanding of the problem. Second, the methodology should ensure nondominated solutions are generated. Finally, the process should not overburden the decision maker (30:719).

1.6 Summary. Multiple criteria optimization techniques provide a method of finding the best compromise solution when a series of objective functions cannot be optimized simultaneously. Interactive methods involve the decision maker in the solution process and are used to solve multiobjective problems where the decision maker's preference structure is unknown or inexpressible.

Vanderpooten classifies interactive multiobjective optimization techniques as search-oriented or learning-oriented (46:1218). Search-oriented methods operate under the assumption that a preference structure exists and is stable. Learning-oriented techniques do not require this assumption but allow the decision maker to express and define a preference structure while searching for the optimal solution.

Whether learning-oriented or search-oriented, the four algorithms studied in this chapter offer variations of the same theme. First, varying methods of obtaining an initial solution to the problem are proposed. Second, different ways of generating additional solutions for review by the decision maker are discussed. Finally, slightly different termination criteria are invoked to signal the arrival of the best solution and to end the process.

There is a general consensus in the literature surveyed that the interactive approach is an ideal multiobjective problem solving technique. Steuer's comment that interactive techniques provide an effective division of labor between the computer and the decision maker is an accurate expression of the underlying theme of interactive multiple objective optimization methodologies (39:361).

2. Location Studies

This section identifies the basic building blocks of location models and relates mathematical programming formulations the types of problems encountered in location studies. The organization and content of this section is borrowed from a paper by Chan and Rowell (11).

2.1 P-Median Problem. The most basic location model is the "simple plant location problem" also known as the uncapacitated facility location problem. This problem needs to be solved to establish a number of facilities with enough capacity to meet all demands. The equations are solved to obtain the lowest cost alternative considering both facility costs and transportation costs. The constraints ensure that all locations are serviced by exactly one facility, and that the selected service facilities are open. The system of equations for this model is as follows:

$$\textit{Minimize} \sum_{j \in I} c_j\, x_{jj} + \sum_{i \in I} \sum_{j \in I} d_{ij}\, x_{ij}$$

subject to:

$$\sum_{j \in I} x_{ij} = 1 \qquad \forall\ i \in I$$

$$x_{jj} - x_{ij} \geq 0 \qquad \forall\ i,j \in I \quad i \neq j$$

$$x_{ij} \geq 0 \qquad \forall\ i,j \in I$$

$$x_{jj} = 0,1 \qquad \forall\ j \in I$$

where
x_{ij} = 1 if location i is serviced by facility j
x_{jj} = 1 if facility j is opened
d_{ij} = cost of servicing location i from facility j
c_j = fixed cost of establishing facility j
I = set of locations for both supply and demand

The "p-median" problem (16) seeks to place p facilities, instead of one facility, among the demands. This type of problem seeks to minimize the average distance or time between facility and servicing locations. The problem can be stated as:

$$\textit{Minimize} \sum_{i \in I} \sum_{j \in I} f_i\, d_{ij}\, x_{ij}$$

subject to:

$$\sum_{j \in I} x_{ij} \geq 1 \qquad \forall\ i \in I$$

$$x_{jj} - x_{ij} \geq 0 \qquad \forall\ i,j \in I \quad i \neq j$$

$$\sum_{j \in I} x_{jj} = p$$

where
f_i = frequency of demand at point i (weight)
d_{ij} = distance from point i to facility j
x_{ij} = 0-1 integer variable

In the above system of equations, the variable x_{ij} is set equal to 1 when facility j is selected to provide services to demand point i. The constraints ensure that all demand points are serviced and that p facilities are selected. Since the weighted distance (or time), d_{ij}, is minimized, the model ensures that each demand point is serviced by the nearest facility.

The p-median problem can be solved using standard linear programming techniques. However, more efficient solutions have been proposed in the literature (6; 3; 40).

If necessary, a constraint can be added to the p-median problem to impose a limit to the distance that must be traveled to reach a facility. This distance is known as the maximum, desirable service distance (43). The additional constraint that must be added to the problem is of the form shown below.

$$\sum_{j \in N_i} x_{ij} \geq 1 \qquad \forall\, i \in I$$

where

$$N_i = \{ j \mid d_{ij} \leq S \} \qquad \forall\, i \in I$$

An interesting application of the maximal, desirable distance constraint is to impose a limit on the worst possible performance (in terms of maximum response time) of the network of facilities. For example, this type of formulation would be useful in locating fire stations where there might be a maximum allowable response time.

2.2 The Set-Covering Problem. Along the same lines as the maximal desirable distance problem, Toregas has proposed a set-covering formulation for facility location problems (42). The set-covering problem seeks to determine the minimum number of facilities such that all users are situated no more than the maximal desirable distance from the service location. Thus, unlike the p-median problem, the number of facilities, p, is a variable instead of a constant. The problem is formulated as follows:

$$\textit{Minimize} \sum_{j \in I} x_{jj}$$

subject to

$$x_{jj} - x_{ij} \geq 0 \qquad \forall\ i,j \in I \quad i \neq j$$

$$\sum_{j \in N_i} x_{ij} \geq 1 \qquad \forall\ i \in I$$

where

$$N_i = \{j \mid d_{ij} \leq S\} \qquad \forall\ i \in I$$

The 0-1 integer variable, x_{jj}, is set to one when facility j is selected. As seen previously, the N_i variable ensures that, for any given demand location, the formulation only considers facilities that are less than the maximal distance. The constraints ensure that all locations are serviced and that the selected facilities are opened.

2.3 The Maximal Coverage Problem. The maximal distance p-median and the set-covering formulations are useful when an unlimited number of facilities can be constructed to meet a minimum demand. When there exists an upper bound on the number of facilities that can be constructed, the models must be modified.

Church and Revelle (12) have proposed a methodology, known as the maximal-covering method, to deal with this type of problem. A second distance, S', is added to the formulation such that S' > S, the maximal service distance. The distance S' represents the maximum distance a given facility can be from any demand location. The constraints ensure that no user is located further than S' away from all facilities, and allows S to be a variable distance ranging between 0 and S'.

The formulation maximizes the population served while meeting the constraint of service distance (d_{ij} < S) and the budgetary constraint (by limiting the number of facilities to p). The service distance, S, can be varied up to the maximum

S', to analyze the tradeoff between travel distance and degree of coverage for varying amounts of p. The problem is formulated as follows:

$$\textit{Maximize } \sum_{j \in I} f_i \, x_{ii}$$

subject to

$$\sum_{j \in I} x_{jj} = p$$

$$\sum_{j \in N_i} x_{ij} \geq x_{ii} \qquad \forall\ i \in I$$

where

$$N_i = \{j \mid d_{ij} \leq S\} \qquad \forall\ i \in I$$

for i demand locations and j facilities

2.4 Probabilistic Methods. The facility location models discussed so far in this paper are for deterministic problems. The data for the problem (distances and demands) are assumed to be constant and known quantities. Whenever any or all of these quantities are random variables, the problem is no longer deterministic but becomes probabilistic in nature.

Mirchandani and Odoni (29) propose the following formulation to deal with p-median problem with k states:

$$\textit{Minimize } \sum_{k=1}^{K} \sum_{i \in I} \sum_{j \in I} Q_k \, f_{ik} \, d_{ijk} \, x_{ijk}$$

subject to:

$$\sum_{j \in I} x_{ijk} \geq 1 \qquad \forall\ i \in I \text{ and } k = 1, \ldots, k$$

$$x_{ijk} \leq x_{jj} \qquad \forall\ i,j \in I; \quad i \neq j; \quad \text{and } k = 1, \ldots, k$$

$$\sum_{j \in I} x_{jj} = p$$

Thus the above formulation seeks to minimize the weighted sum of the travel distance over all possible states. The probability associated with each state, Q_k, provides the weight. The variable x_{ijk} takes on the value of one when demand point i is assigned to facility j in state k. Therefore, this type of formulation allows for facility assignments to vary from state to state based on the value of d_{ijk} which could represent, for example, varying travel times based on time of day.

2.5 Stochastic Facility Location. Odoni advances that there are two types of uncertainties in facility location problems (34):

(1) random travel times along the arcs of the network

(2) queuing phenomena arising from a combination of finite capacity at the facilities and random location of demands on the network, or random arrival time of the demands and random service times.

When the uncertainty is caused by random travel times, the problem can be solved using the deterministic p-median formulation as described above. However, when uncertainties are due to the queuing phenomena, "research efforts have been focused upon single-facility problems due to the severe analytical difficulties associated with multi-facility problems" (12).

The problem in this research paper could potentially involve uncertainties of the second type. The satellites will have pseudo-random arrival times and require random service times. Queuing, in the strict sense of the word, will not apply to orbiting satellites. However, the demand for observations generated by operational tasking requirements will need to be met in a timely manner.

2.6 Location on a Plane and Network. There are three ways to measure distances in facility location problems: the Euclidean metric, the Manhattan metric, and the continuous median method.

The Euclidean metric method utilizes the straight-line distance between points. For example, this type of measurement would apply to air travel where users can travel directly from the demand point to the facility. The Manhattan metric technique restricts travel along grid lines parallel to the coordinate axis. An obvious application is when travel between two points must follow the rectangular pattern of streets in a city (12).

In a network, the continuous median "is a point such that the sum of the distances between all arcs and the point is minimum" (12). The distance between an arc and a point is the length of the longest line that can be drawn from the point to the arc. Similarly, the continuous p-median is a set of p points such that the sum of the distances from all arcs to the closest point to each arc is a minimum.

2.7 Summary. The basic single facility location model is readily extended to the p-median problem where p facilities are optimally located to satisfy demands. This type of problem minimizes the average distance or time between a facility and the locations it services. A constraint can be added to the p-median problem to limit the distance from a demand point to a facility. This distance is known as the maximum desirable service distance.

The set-covering formulation is similar to the maximal desirable distance problem formulation. Both methodologies locate the minimum number of facilities required to serve the demand points such that all users are situated no further away than the maximal distance. The maximal coverage technique has been developed to deal with problems where there is a limited number of available facilities.

Probabilistic methods provide ways of dealing with problem parameters that are random variables. Mirchandani and Odoni (29) have proposed a formulation to deal with p-median problems with multiple states.

Odoni (34) has identified two types of uncertainties in facility location problems. Problem with uncertainties due to random travel times along the arcs can be solved using deterministic p-median formulations. Other types of uncertainties may create severe analytical difficulties.

Three ways to measure distances in facility location problems have been identified: the Euclidean metric, the Manhattan metric, and the continuous median methods.

3. Network-Flow Programming

Most of us can readily grasp the concept of a network model when we see the simple diagram of a number of arcs connecting together a number of nodes (or vertices). Physical significance can immediately be attached to the elements of a network model as one sees the parallel with a communication network or visualizes the flow of fluids through pipes (the arcs) and pumping stations (the nodes).

Network modeling techniques can be applied to several types of problems, and are applied in this research to two facility location problems (Chapter IV). Representing these problems as networks helped provide an understanding of the dynamics of the problem. Phillips and Garcia-Diaz list the following advantages of using network models (35):

1. Network models accurately represent many real-world systems.
2. Network models seem to be more readily accepted by nonanalysts than perhaps any other type of models used in operations research. This phenomenon appears to stem from the notion that "a picture is worth a thousand words"...
3. Network algorithms facilitate extremely efficient solutions to some large-scale models.
4. Network algorithms can often solve problems with significantly more variables than can be solved by other optimization techniques. This phenomenon is due to the fact that a network approach often allows the exploitation of particular structures in a model.

3.1 Definitions and Terminology. The following definitions are taken from Chan (10) and Phillips and Garcia (35).

3.1.1 Graph. A graph, G(V,E), is a set of nodes V={1,2,...,m) connected by edges $E=\{e_1,e_2,...,e_n\}$.

3.1.2 Network. A network is a graph with flow of some kind.

3.1.3 Unimodular. A square, integer matrix is called unimodular if its determinant, det(B) = +/- 1. An integer matrix A is called totally unimodular (TU) if every square, non-singular submatrix of A is unimodular.

3.1.4 Source/Sink. A source in a network is a node where units of flow enter the network. Conversely, a sink in a network is a node where units of flow leave the network. Networks can be designed with multiple sources and sinks.

3.1.5 Pure/Generalized Networks. In a pure network, there are no losses or gains of units of flow through the network. For every unit of flow entering the network, there is one unit of flow leaving the network. In a generalized network, losses or gains can be modelled to occur at nodes and/or arcs. The flow in and/or out of the network at a given node can be fixed or variable.

3.2 Theorems.

3.2.1 If matrix A is TU, then all the vertices of the polyhedron $P(b) = \{ x \in R^n_+ : Ax \leq b \}$, are integer for any integer vector b, $b \in Z^m$. (i.e., an integer solution can be obtained to a linear program without the need to impose integer restriction to the variables when the constraint matrix A is TU).

3.2.2 An integer matrix A with all elements a_{ij}=0, +/- 1, is TU if no more than two nonzero entries appear in any column, and if the rows of A can be partitioned into two sets, Q_1 and Q_2 such that:

(1) If a column has two entries of the same sign, their rows are in different sets Q_1 and Q_2, and

(2) If a column has two entries of different signs, their rows are in the same set Q_1 and Q_2.

4. Climatology

4.1 The Canadian Climates. The Canadian Encyclopedia (41:354) lists 5 main climatic regions for the southern populated areas of Canada: East Coast, Great Lakes, Prairies, Cordilleran (Rocky Mountain) and West Coast. Further, the authors claim that while many different climatic regions exist in the far North, the mostly uninhabited area of northern Canada can be subdivided into Arctic and Subarctic climatic regions.

The subdivisions found in the Canadian Encyclopedia closely parallel those presented by Trewartha et al. (45) who identify a "Polar Tundra" region for the Arctic, a "Boreal" region for the Subarctic, a "Temperate Oceanic" region for the West Coast, and a "Highland" region for the Cordilleran. Trewartha et al. group most of the Prairies, and the Great Lakes and East Coast regions under the heading of "Temperate Continental" and identifies a portion of southern Saskatchewan and Alberta as belonging to the "Dry Steppe" climate group.

"Defining climatic regions for any country is difficult... Within a geographical area, climates gradually change from one type to another" (41:353). For the purposes of this research, the climate regions identified in the Canadian Encyclopedia will be adopted. The following climate types "result from the relationship between monthly potential evapotranspiration, PE, (or need for water) and precipitation" (41,354):

> **East Coast.** These climates are represented by Halifax, NS. Precipitation is fairly uniform throughout the year and only in July does water need exceed supply
> **Great Lakes.** These Southern Ontario climates are typified by Windsor. Precipitation is rather uniform throughout the year ...
> **Prairie.** These climates are exemplified by Edmonton. The annual precipitation is inadequate to meet the PE and deficits are common during the summer months. With low winter precipitation, soil moisture is not always restored...

Cordilleran. This region is a composite of many climatic types. The southern BC valleys have climates that are the driest in Canada.
West Coast. These climates are characterized by a winter maximum and summer minimum precipitation regime. Victoria is a typical station.
Subarctic and Arctic. At Inuvik, the monthly PE and precipitation resemble those of stations in the Prairies. Alert, in the High Arctic, has a more severe arctic climate, with low precipitation.

4.2 Determining Cloud-Free-Line-Of-Sight (CFLOS). To obtain useful data, the GEODSS must acquire and maintain a CFLOS on a satellite for a period of five minutes (19). The model developed in this research requires the probability of having a CFLOS from a given location when observing a given satellite. Methodologies to determine CFLOS probabilities were first proposed by Lund (24) and McCabe(27) and much has been written on the subject since that time (47).

The CFLOS probabilities required for this research were provided by the USAF Environmental Technical Applications Center (ETAC/DNY) located at Scott AFB. The analytical model used by ETAC/DNY to determine the CFLOS probabilities is based on the work of Malick et al. (25) which represents the current standard in the field (25).

There are two distinguishing feature to the Malick et al model (25:142). First, the probability calculations are performed analytically whereas previous efforts had been largely empirical. Second, the concept of CFLOS was generalized to determining the probability of a cloud-free interval which includes the probability of CFLOS as a special case. "This methodology can be used to calculate the probability of a cloud-free interval of various lengths (of time) within a straight line path of any length" (25:142).

The ETAC/DNY model requires sky cover climatology derived from ground observations as input to produce the CFLOS probabilities for a given location and satellite (47). Therefore, the availability of historical weather data becomes a factor in selecting candidate sites to locate a GEODSS station. This is not considered a serious limitation of the methodology proposed in this research since, from a practical point of view, it would not be advisable to select a location without first reviewing historical weather patterns associated with the proposed site.

5. Transmissivity of the Atmosphere

To acquire and track satellites, the GEODSS telescopes must gather sunlight which is reflected by the satellites they observe. What is commonly referred to as "light" is more technically labeled as electromagnetic radiation (EMR) with wavelength of 4-7 μm. If an electro-optical device such as GEODSS is to perform its function adequately, then its detector must be able to gather sufficient amounts of EMR to recognize the presence of a target in its field of view.

The interaction of the atmosphere with the reflected EMR traveling from the satellite to the GEODSS telescope causes a fraction of the energy to be either absorbed or scattered. The end result is that only a portion of the available reflected EMR actually reaches the GEODSS telescope. It is important to understand these interaction mechanisms as they will obviously affect the decision of where to locate a GEODSS observing station.

The Manual of Remote Sensing published by the American Society of Photogrammetry (1) provides, among other topics, an excellent description of the nature of electromagnetic radiation and of the interaction mechanisms within the atmosphere. This information is sketched out here for the convenience of the reader.

5.1 Scattering and Absorption. As already mentioned, the transmissivity of the atmosphere is defined by the amount of scattering and absorption of the EMR. Transmissivity of the atmosphere varies with the wavelength of the EMR (1:181).

The atmosphere consists of particles of various sizes ranging from particles of atomic and molecular size to larger particles (or particulates) with radius of up to

100 μm (1:187). All of these particles can scatter the EMR. However, scattering from atomic size particles is negligible and molecular scattering is important only in the near ultraviolet and visible part of the EMR (1:181). Unfortunately, this happens to be the area of interest for GEODSS and other electro-optical devices.

The amount of molecular scattering and absorption that takes place is not only a function of wavelength but a function of the density of the atmosphere. In other words, the chances of interaction between EMR and a particle in the atmosphere increases when there are more particles present. The density the atmospheric gases (i.e., the number of molecular size particles) decreases at an exponential rate with respect to altitude above sea level (1:181). Therefore, to maximize the amount of reflected EMR received from a satellite (or minimize the losses through the atmosphere) it makes sense to want to locate an observing station at higher elevations.

The larger size particles, or particulates, are both natural and man-made in origin. Only particulates between 0.06 and 10 μm in diameter have an effect on the transmission of EMR in the visible wavelengths (1:187). About 60% of these particles are concentrated near the surface of the earth, within 1 km of the ground. Sources of natural particulates smaller than 20 μm in radius include soil debris, forest fires, and volcanic eruptions. Man-made sources are primarily as a result of gaseous emissions (1:186).

5.2 Ambient Light. Experience tells us that it is much easier for the naked eye to detect a distant light source at night than in the daytime. The same holds true for the GEODSS detector. In fact, the GEODSS will only operate when the sun is at least 6 degrees below the local horizontal plane (19). This is due to the fact that the presence of a target can only be detected if the source of EMR is distinguishable from the background EMR. Thus, an observing site should not be located near sources of EMR such as cities, which will impair the ability of the detector to detect the target.

6. Orbital Mechanics

6.1 Classical Orbital Elements. The size, shape, and orientation of an orbit can be completely described by five independent quantities called "orbital elements". Given a sixth element, the position of a satellite in the prescribed orbit can be determined at any moment in time. These six elements are known as (1) the semi-major axis, (2) the eccentricity, (3) the inclination, (4) the longitude of the ascending node (or the right ascension of the ascending node), (5) the argument of periapsis, and (6) the time of peripasis passage. Together, these six parameters form the classical set of orbital elements and are defined as follows (4:58):

(1) The semi-major axis is a constant which defines the size of the conic orbit.

(2) The eccentricity is a constant which defines the shape of the conic orbit.

(3) The inclination, in the case of earth-orbiting satellites, is the angle made between the earth's rotational axis and a vector perpendicular to the orbital plane known as the angular momentum vector (a more technically precise definition can be found in Bate et al. (4:58)).

(4) The longitude of the ascending node, in the case of earth-orbiting satellites, is an angle measured in the equatorial plane, from the first point of Aries to the point where the satellite in its orbit crosses the equatorial plane in a northerly direction (ascending node). The angle is measured counterclockwise when viewed from the north side of the equatorial plane.

(5) The argument of periapsis, which in the case of earth-orbiting satellites is called the argument of perigee, is an angle measured in the orbital plane from the

ascending node to the perigee point (the point where the satellite is closest to the earth) in the direction of satellite travel.

(6) The time of periapsis passage, which in the case of earth orbiting satellites is called the time of perigee passage, is the time when the satellite was at perigee.

While not one of the six classical orbital elements, another important orbital parameter is the period which equates to the time taken by the satellite to complete one full revolution in its orbit.

6.2 Common Orbit Types.

6.2.1 *Geostationary.* A geostationary orbit is a circular orbit with an inclination of zero (or near zero) degrees, and with a period of one sidereal day. Satellites in geostationary orbits that are traveling in the same direction as the earth's rotation appear stationary with respect to the earth.

6.2.2 *Earth Synchronous.* A satellite which completes an integral number of orbits per day is said to be earth synchronous (37:46). An example is the Molniya (also spelled Molnya and Molnia) which is called a semi-synchronous orbit because it has a one-half sidereal day period. The Molniya is an orbit used by the Soviet Union for telecommunications purposes (37:44).

6.2.3 *Sun Synchronous.* The distinguishing feature of a sun synchronous orbit is that the orientation of the orbital plane remains fixed with respect to the sun. For example, the orbital plane may be kept facing the sun which provides continuous illumination of the solar panels for the life of the satellite. Fixed orientation with respect to the sun is achieved by allowing the orbital plane to precess at a rate of 0.9856 degrees-east per day (37:45). The precession of the orbital plane is caused by the oblateness of the earth, and the rate of precession is a function of the semi-major axis, inclination, and eccentricity parameters of the orbit (37:23).

6.3 Orbit Determination from Optical Sightings. Bate et al. provide an excellent exposé of a solution methodology to the problem of solving for the classical orbital elements given optical sightings alone (4:117).

Bate et al. explain that the problem of solving for the classical elements is much simpler when radar information, including range and range-rate, is available. However, they point out that the angular pointing accuracy of radars is much less than that available from optical instruments and that much more precise determinations of orbital parameters is therefore possible from optical sightings (4:117).

Bate et al. also explain that since six independent quantities are required to fully describe an orbit and to position a satellite in that orbit, that a minimum of three optical sightings taken at three different times are required to solve the problem. This is due to the fact that each optical sighting provides two independent quantities (such as azimuth and elevation) and thus three observations provides for six equations with six unknowns. The equations used to solve the problem are fully derived by Bate et al., together with a differential correction technique to make use of redundancy of observation to improve the solution (4:122).

6.4 Determining the Limits of Visibility. Only a segment of the geostationary belt is visible from a given geographic location. One way to identify these limits of visibility is in terms of the longitudes of the satellite subpoints.

Roddy provides a very straightforward technique to determining the limits of visibility of the geostationary belt from a given location on earth (37:52). His development adopts the following notation and conventions:

Φ_E = longitude of the earth station, where longitudes east of the Greenwich meridian are taken as positive numbers (the greek symbol "phi" used by Roddy for longitude commonly represents latitude in other literature)

Φ_S = longitude of the satellite subpoint

λ_E = latitude of the earth station, where north latitudes are taken as positive numbers

λ_S = latitude of the satellite subpoint

R_E = earth equatorial radius (6378.14 km)

R = earth radius as a function of latitude

h = geostation-ry height (35,786 km)

S = angle subtended at the satellite in a triangle formed by joining the center of the earth, the geographic location, and the satellite position

c = spherical angle measured from the geographic location to the satellite subpoint in a spherical triangle formed by joining the north pole, the geographic location and the satellite subpoint

B = interior angle of a spherical triangle representing the difference in longitude between the observer location and the satellite subpoint. The spherical triangle is formed by joining a pole with the earth station location and the satellite sub-point.

The algorithm to solve for the limits of visibility is as follows (37:52):

$$R = R_E \left(1 - \frac{\sin^2\lambda_E}{298.257}\right)$$

$$\sin S = \frac{R}{R_E + h} \sin El$$

$$c = 180 - (El + S)$$

$$\cos B = \frac{\cos c}{\cos \lambda_E}$$

$$Limits = \phi_E \pm B$$

Roddy also provides a very straightforward solution algorithm to determine the "look angles" to a geostationary satellite from a given earth location. A look angle is the angular orientation assigned to a satellite tracking instrument in order to locate a satellite as it orbits the earth. The equations in this algorithm are based on the same parameters used in the above limits of visibility calculation (37:48).

III. METHODOLOGY

This chapter describes the solution methodology that is developed in this research paper. The solution methodology is graphically portrayed at Figure 1. The main headings of this chapter parallel the list of sub-objectives that were presented in Chapter I.

1. The Set of Candidate Locations

The candidate locations must be chosen from a list of locations within Canada for which airfield weather summary information is available. This is necessary because the probability of favorable weather for a given location will be determined using airfield weather data.

The number of candidate locations must be kept to a minimum. As discussed below, the problem of finding the optimal location is combinatorial, with an explosive number of feasible alternatives given by,

$$\frac{n!}{(n-p)!\,p!}$$

$$\begin{aligned} \textit{where } p &= \textit{number of facilities} \\ n &= \textit{number of locations} \end{aligned}$$

The set of candidate locations must be dispersed as evenly as possible throughout Canada to take into account the impact of geographic location on the number of observations that can be collected on a satellite in a given orbital plane. A cross-reference to climatological maps ensures that the candidate location set also provides representatives from each of the Canadian climatic regions identified in Chapter II.

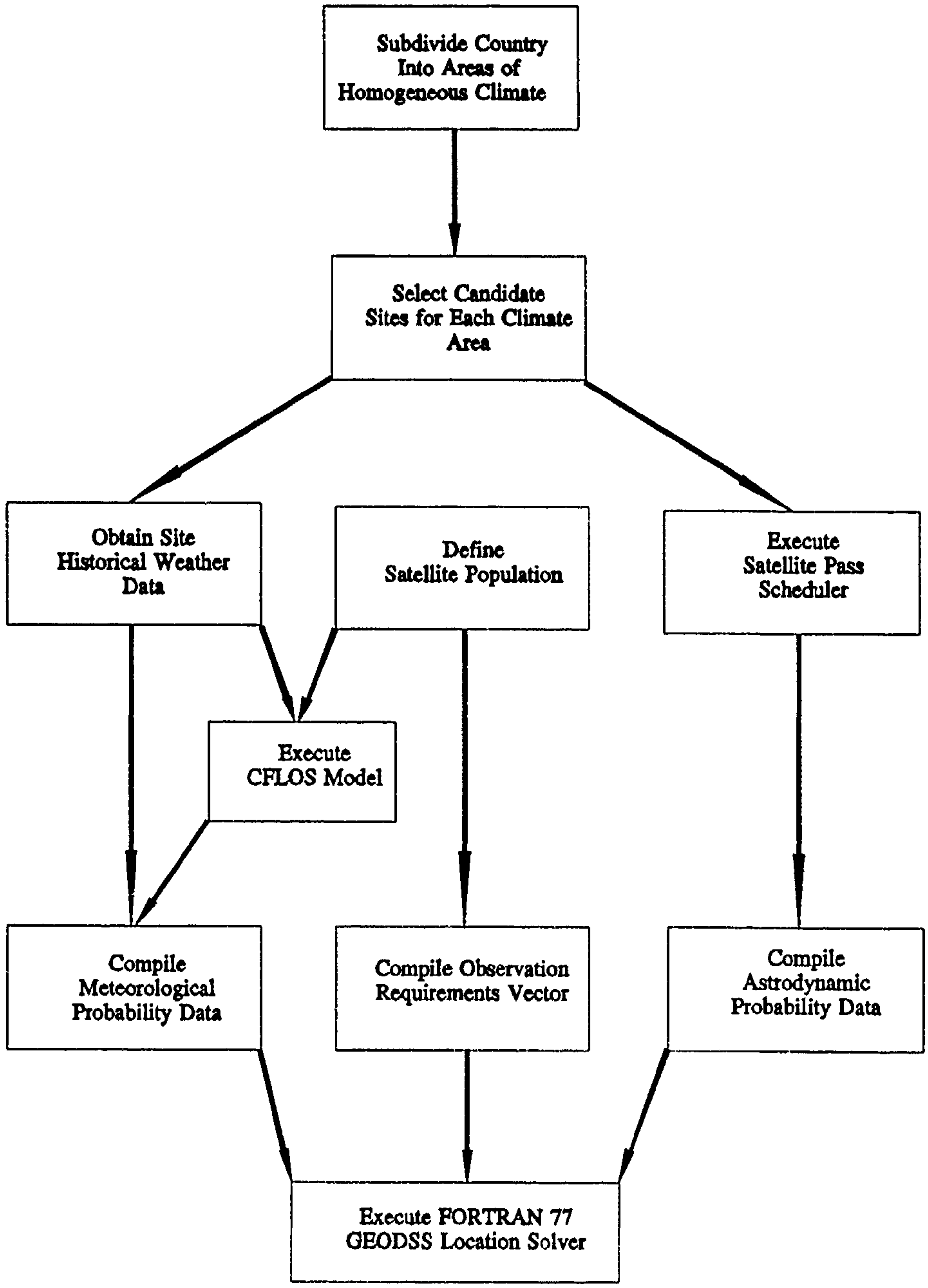


Figure 1 - Methodology

As discussed below in Section 4, the criteria set in this research is limited to two. These two criteria distinguish themselves from all other criteria identified through discussions with the decision maker in that they have a "more is better" quality. All other criteria can be treated as pass/fail, or essential criteria. These include site accessibility, availability of personnel support resources (such as housing), ambient lighting, and others. All locations chosen as candidates must, of course, meet the essential criteria.

2. The Probability a Time Block is Observable

The GEODSS system operating specifications in terms of environmental factors were provided by NDHQ (18). In particular, cloud-free-line-of-sight, temperature, and wind speed criteria that restrict the operation of the GEODSS system were identified and are addressed in this research. In addition to suitable environmental conditions, the satellite must remain in view for 5 minutes, be at least 15 degrees above the horizontal plane, and be illuminated by the sun during conditions of civil twilight (i.e., when the sun is at least 6 degrees below the horizon).

Given all of the above, the probability a time block of five minutes is usable for observing is given by the following equation (31):

$$P(ABCDEF) = P(A)\ P(B|A)\ P(C|AB)\ P(D|ABC)\ P(E|ABCD)\ P(F|ABCDE)$$

where

A = sun < 6 degrees below horizon
B = wind speed < 25 knots
C = temperature > -50 C
D = satellite > 15 degrees above horizon
E = CFLOS for 5 minutes
F = satellite illuminated by the sun
AB = intersection of events A and B

The following assumptions of independence of events are made:

(1) Event D is independent of ABC which implies that P(D|ABC)=P(D)

(2) Event F is independent of BCE which implies that P(F|ABCDE)=P(F|AD) (32)

The above probabilities are obtained for each candidate location, and for every month of the year, to take into account seasonal variations of the climatological parameters. The probabilities, P(A), P(B|A), P(C|AB), P(D), and P(E|ABCD) are provided by ETAC/DNY, Scott AFB (48).

The probability, P(F|AD), is derived using data from AFIT/ENS software developed by Dr T.S. Kelso which, among other options, determines when events FAD and AD occur during a given time period, geographic location, and for a given satellite (22). The total number of minutes in a month when events FAD and AD occurred is divided by the total number of minutes in a month to obtain P(FAD) and P(AD) respectively. These probabilities are then used to calculate P(F|AD) = P(FAD)/P(AD).

Alternatively, occurrence of events FAD and AD could be obtained from a software package by the name of SATRAK which is available from AFSPACECOM/DOJ (17).

3. The Expected Number of Observable Time Blocks

The expected number of usable observation opportunities for each candidate location for every month of the year will be calculated using the equation

$$E_{ijk} = P_{ijk} B_j$$

where

E_{ijk} = expected number of observation opportunities in month 'j' at location 'i' of satellite 'k'
P_{ijk} = the probability block is observable
B_j = the total number of 5 minute observation blocks month j

In this research, the expected number of observation opportunities, E_{ijk}, is the fundamental building block of the optimization process. The effectiveness of an alternative is judged in terms of the total number of observation opportunities and also in terms of the variance in the number of observations. Both of these quantities are calculated from E_{ijk}.

4. Multiple-Criteria Facility Location Model

The sponsor for this research asks that the optimal one-site, two-site, and three-site configurations be determined (18). While this question is directly translated into a maximal coverage facility location problem, a set covering facility location formulation offers a slightly different and interesting way to approach the solution. Thus, single objective maximal coverage and set covering models are developed and presented in this research. The final multiple-criteria facility location model includes elements of both types of models.

4.1 Criteria Functions. The two criteria functions are given by:

$$Max\ f_1 = \sum_{i}^{I} \sum_{j}^{J} \sum_{k}^{K} E_{ijk}\ X_{ijk}$$

$$Min\ f_2 = \sum_{j}^{J} V_j$$

The first criterion function seeks to maximize the total expected number of observation opportunities. The second criterion function seeks to minimize the sum of the variances, V_j, in the number of observations collected on a given satellite in a given month, by an alternative. Thus, the variance function seeks to ensure that each satellite is equally well observed in a given month.

4.2 Model Constraint. The problem is formulated with only one constraint. A minimum number of observations must be collected in any given month on a given satellite to ensure that the constantly changing orbital parameters are adequately maintained. This constraint is stated mathematically as follows:

$$\sum_{i}^{I} E_{ijk} X_{ijk} \geq D_k \qquad \forall j,k$$

The vector, D_k, represents the monthly "quotas" of observations required on each satellite "k" to maintain the necessary update frequency of the orbital parameters. The requirement D_k is provided by Headquarters 1st Space Wing (26).

4.3 Solution Algorithm. The set of candidate locations is restricted to 12. This number is considered sufficient to provide representatives of all major Canadian climatic regions while at the same time providing full geographic coverage for reasons stated above. The satellite target population is limited to 6. This latter restriction is not a function of the software and the number of satellites could be increased if the probability data described above is available.

Using the formula presented in Section 1 above, and given a set of 12 locations, there are 12 one-site, 66 two-site, and 220 three-site alternatives, for a total of 298 alternatives. First, E_{ijk} is computed from probability tables as discussed above. Feasible alternatives can then be identified by testing each alternative against the constraint equation. Next, the values of both criterion function are computed for all feasible alternatives providing a complete description of the outcome space for each of the three subproblems: the one-site, two-site, and three-site problems.

Since the problems are integer programming, the complete set of N-points cannot be generated using linear programming techniques such as MC-simplex (31:219). Instead, the complete set of N-points, as defined with respect to Pareto preference (50:10), is extracted from the set of feasible outcomes by verifying dominance of each point against all others.

The solution procedure concludes by ordering the alternatives based on deviation from the ideal for each of the three subproblems. The ideal is defined as a point (y_1^*, y_2^*) in the outcome space with $y_1^* = \{\max y_1^i\}$, and $y_2^* = \{\max y_2^i\}$, given "i" feasible alternatives. Ordering of alternatives is obtained by sorting the N-points in ascending order of Manhattan distance to the ideal.

From the ordered list, the optimal alternatives for the maximal coverage p=1, p=2, and p=3 subproblems are simply the leading candidates of each list. The optimal alternative for the setcovering problem would be the top alternative of the p=1 list. If there are no feasible alternatives for the p=1 problem, then the set covering solution is the top alternative for the list which includes feasible alternatives and has the lowest p value.

4.4 Software Package. The solution algorithm described above is programmed in FORTRAN 77. The source file for this software is provided at Appendix U and is documented in Chapter V.

The input to the program consists of seven data files containing the probabilities associated with the events described earlier in this chapter:

PROBA.DAT	12x12 matrix of probability of event A at location I, in month J
PROBB.DAT	12x12 matrix of probability of event B given A, at location I, in month J
PROBC.DAT	12X12 matrix of probability of event C given AB, at location I, in month J
PROBD.DAT	12x6 matrix of probability of event D, at location I, for satellite K
PROBE.DAT	six 12x12 matrices of probability of event E given ABCD, at location I, in month J, for each satellite
PROBF.DAT	six 12x12 matrices of probability of event F given AD, at location I, in month J, for each satellite
OBSREQ.DAT	1x6 vector of monthly requirement of observations for each satellite

The output also consists of eight files as follows:

RESULTS.OUT	three tables of top 12 alternatives ordered by deviation from the ideal, and a table of E_{ij} for each of the six satellites.
ALTLST.OUT	table showing alternative numbering legend used by the software
ALTOBS.OUT	six 298x12 tables showing number of observations collected by alternative X, in month J, on each satellite
OBJFCN.OUT	table showing valu^ of F_1 and F_2 for each alternative
UTILS.OUT	table showing utility of F_1 and F_2 for each alternative
FEASIB.OUT	table showing the feasibility status (true or false) of each alternative
EFFSET.OUT	table indicating whether or not an alternative is part of the non-dominated set
DEVIAT.OUT	table showing the Manhattan metric deviation from the ideal for each alternative

IV. SINGLE OBJECTIVE MODELS

1. Maximal Coverage Formulations

As was seen in Chapter II, the classical "p-median" facility location problem seeks to place p facilities among the demands. This type of problem seeks to minimize the average distance or time between facility and servicing locations. The problem is re-stated here for the convenience of the reader:

$$\text{minimize} \sum_{i \in I} \sum_{j \in I} f_i \, d_{ij} \, x_{ij}$$

subject to:

$$\sum_{j \in I} x_{ij} \geq 1 \qquad \forall\, i \in I$$

$$x_{jj} - x_{ij} \geq 0 \qquad \forall\, i,j \in I \quad i \neq j$$

$$\sum_{j \in I} x_{jj} = p$$

where

f_i = frequency of demand at point i (weight)
d_{ij} = distance from point i to facility j
x_{ij} = 0-1 integer variable

In the above system of equations, the variable x_{ij} is set equal to 1 when facility j is selected to provide services to demand point i. The constraints ensure that all demand points are serviced and that p facilities are selected. Since the weighted distance (or time), d_{ij}, is minimized, the model ensures that each demand point is serviced by the nearest facility.

1.1 Mathematical Formulation. The classical p-median model is readily adapted to the GEODSS location problem. The objective function is modified to maximize the number of observation opportunities, W_{ijk}:

$$\text{maximize} \sum_{i}^{I} \sum_{j}^{J} \sum_{k}^{K} W_{ijk} X_{ijk}$$

$$\text{where } X_{ijk} \text{ is } 0\text{-}1 \text{ integer}$$

The observation opportunities for location i, in month j, of a type k satellite are obtained by multiplying the probability of favorable weather times the total number of observation for that month. An observation opportunity occurs when a satellite is in the field-of-view of the telescope for a period of five minutes, is above 15 degrees in elevation, and is illuminated by the sun. The total number of such opportunities for a given month can be determined from celestial mechanics. For the purposes of this project, artificial data will be used.

The following three constraints ensure that, (1) the selected facility is open, (2) that 'p' facilities are selected, and (3) that a minimum monthly observation quota is met for each type of satellite:

$$X_i - X_{ijk} \geq 0 \qquad \forall\, i,j,k$$

$$\sum_{i}^{I} X_i = p$$

$$\sum_{i}^{I} W_{ijk} X_{ijk} \geq d_k \qquad \forall\, j,k$$

The number of locations, i, under consideration, the number of states (months), j, and the number of satellite types, k, will be limited to three for the purposes of this project. However, approximately 10 locations and 10 satellite orbit classes need to be included in the actual problem, resulting in at least 1200, 0-1 integer variables (obtained by multiplying ixjxk). A search for alternate solution methods is proposed.

1.2 Network-Flow Formulation. Various approaches were attempted to convert the above mathematical program into a network flow formulation. Efforts to obtain a pure network flow model did not prove successful. These efforts were motivated by a desire to take advantage of the totally unimodular properties of this type of formulation to increase computational efficiency. However, it eventually became clear that a generalized network, or network with gains, was required.

The network diagram is shown at Figure 2. As stated previously, the formulation is limited to i=3 candidate facilities, to j=3 states, and k=3 demand points. Nodes 1, 4, and 7 represent facility 1 in states 1, 2, and 3 respectively. Similarly, nodes 2, 5, and 8 represent facility 2, and nodes 3, 6, and 9 represent facility 3. Also, nodes 10, 13, and 16 correspond to demand point 1, nodes 11, 14, and 17, correspond to demand point 2, and nodes 12, 15, and 18, to demand point 3.

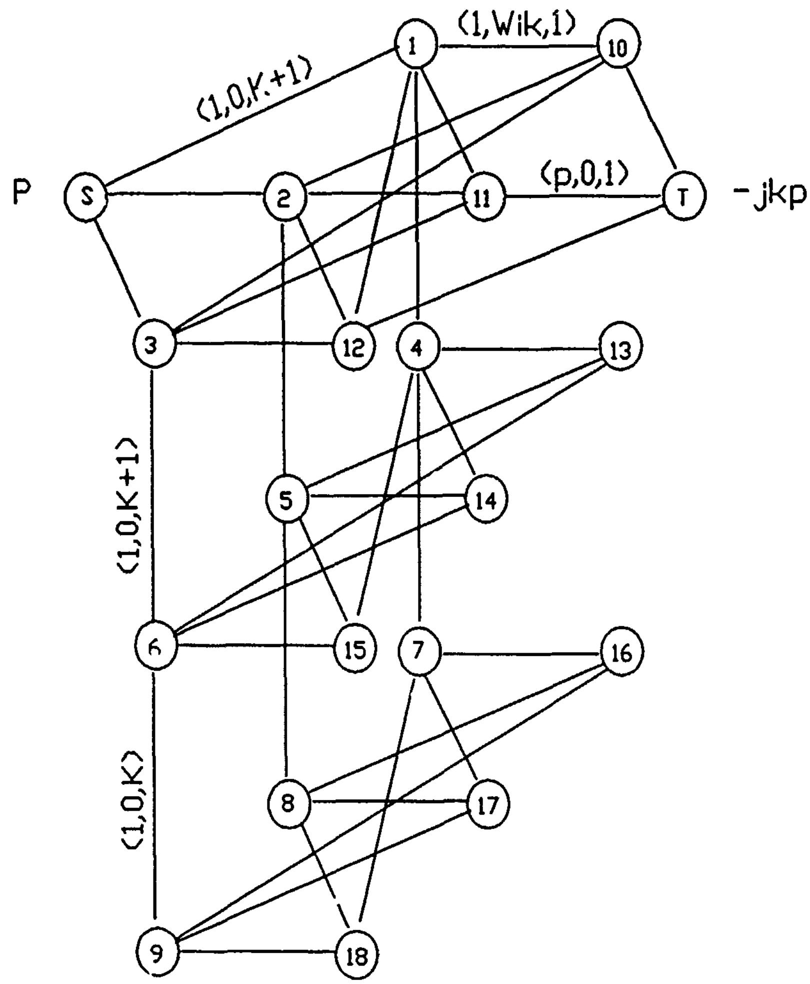
i\k	10	11	12	TOTAL
1	8	5	4	17
2	3	6	9	18
3	2	7	5	14
TOTAL	13	18	18	49

Table 2 - W_{ik} for State 1

i\k	13	14	15	TOTAL
4	6	2	7	15
5	9	3	6	18
6	4	6	5	15
TOTAL	19	11	18	48

Table 3 - W_{ik} for State 2

The fixed external flow at the source is set equal to the number of facilities 'p' to be selected. Arcs with capacity of 1, zero weight, and gain of k+1, connect the



NOTES: (1) Arcs connecting nodes 13 to 18 to node T (sink) are not shown

(2) (u,w,a) = (upper bound, weight, and gain)

Figure 2 - Network Flow Diagram (Maximal Coverage)

i\k	16	17	18	TOTAL
7	4	8	9	21
8	6	7	9	22
9	3	9	7	19
TOTAL	13	24	25	62

Table 4 - W_{ik} for State 3

source with the facilities in state 1 only. Within each state, arcs link a given facility with all three demand points. The capacity of these arcs is 1 and the weights, w_{ik}, are as shown in Tables 2, 3, and 4. "Interstate" arcs, with capacity 1, weight 0, and gain k+1 (except for the arcs between states j-1 and j which have a gain of 'k') connect the nodes representing a given facility. Arcs connecting demand point nodes to the sink are assigned a capacity of 'p' and weight 0. The fixed external flow at the sink is set equal to (-jkp).

The total number of nodes in the network for a problem with 'i' locations, 'j' states, and 'k' demand points, is simply the 2 (source and sink) plus the number of nodes in each state multiplied by the number of states: (2 + j(i+k)). The total number of arcs is the sum of, (1) the number of arcs out of the source, (2) the number of arcs into the sink, (3) the number of arcs connecting the facilities and demand points multiplied by the number of states, and (4) the number of "interstate" arcs. This reduces to j(i+k+ik).

Thus, for the limited version of the problem, with i=j=k=3, there are 20 nodes and 45 arcs. The network representing the actual problem, which as stated previously should include i=10 locations, j=12 months, and k=10 satellite orbit classes, will therefore consist of 242 nodes and 1,440 arcs.

1.3 Solution Procedure Using MIP83. The above network-flow formulation leaves out the minimum monthly observation constraint of the original mathematical formulation. This constraint, which should be added as a side constraint, is ignored for the maximal coverage formulation so that the MIP and Branch and Bound solution procedures adopted for this research can be compared.

The purpose of the minimum monthly observation constraint is to ensure that a minimum number of observations is obtained each month for each satellite type. Leaving out this constraint means that the optimal location(s) will be the one(s) which provide(s) the greatest total throughput potential. It is conceivable that for any given state (month), the optimal choice selected on the basis of this rule may not provide the minimum requisite number of observations.

The formulation will only produce the desired output for integer flow. Unfortunately, because of the presence of gains, the node-arc incidence tableau is not totally unimodular and an integer solution is not guaranteed. This was verified using MICROSOLVE/OR and MICROSOLVE/Network Flow Programming which both produced non-integer flow solutions (20; 21). Introduction of the gain parameter also increases solution time and memory requirements of the Microsolve programs (20:40).

To force an integer solution on the system of equations, the MIP/83 software package was used to solve the problem (23). This package is very "user friendly" only requiring an ASCII text file with .LP extension as input. The output can be directed to an ASCII text file. The program is accessed from the directory containing the MIP83.EXE file, by typing the following command line:

MIP83 filename.LP OUTPUT filename

The command line can include several other job control parameters other than the OUTPUT command. Command line parameters are outlined in Section 5-1 of the user's manual.

The input file is included at the beginning of the output file. The structure of the input file is simple. As shown at Appendix A for example, each file must begin with the ..TITLE statement and be followed by a title. The ..OBJECTIVE section follows the ..TITLE section, indicating whether this is a max/min problem and spelling out the objective function. This may be followed by a ..BOUNDS section if required to set lower and upper bounds on the variables. The ..CONSTRAINT section completes the package. It is important to note that all variables must be identified in the objective function even if they are zero cost. Also as a means of identification, in the objective function integer variables are single-square bracketed, and 0-1 integer variables are double-square bracketed.

An excellent feature of MIP83 is the provision of comment lines in the input file. These are identified by using an asterisk as the leading character. This allows the user to include documentation with the input file. Also, MIP83 allows the constraint to be labeled if desired by preceding the constraint equation with the label name and a colon. The output file at Appendix A shows the use of both comment lines and constraint labels in the input section.

MIP83 provides excellent error handling. Coded diagnostic messages are generated. A complete list of error messages and their reference number is provided at section 9 of the user's manual.

The output based on the above data for p=1 and p=2 is shown at Appendix A and B. As shown in these annex, MIP83 includes a tabular representation of the problem variables with their respective value at optimality together with the objective function coefficient associated with each variable. Also included in tabular form is the value of the left-hand and right-hand sides of each constraint of the problem.

By inspection of Tables 2 through 4, the optimal solution of selecting site 2 for p=1 (objective=58) and selecting sites 1 and 2 for p=2 (objective=111) is correctly identified by MIP83. The optimal integer solution and the optimal solution to the relaxed problem are shown at the head of the tables.

1.4 Solution Procedure Using LP83. As indicated above, the node-arc incidence matrix of the network is not totally unimodular and an integer solution is not guaranteed. An attempt was made to force an integer solution on the problem by adding a series of side constraints that impose conditions on the flow through the network.

Appendix C shows the LP83 (23) output of the resulting formulation. From a user interface perspective, LP83 is identical to MIP83. The only significant difference is that the variables in the objective function are not single or double-bracketed to impose integer restrictions.

As can be seen at Appendix C and D, three types of constraints were added to the MIP problem. First, the "source-connector flow" constraints ensure that the upper bound on the flow through each of the "source-connector" arcs is limited to 1. Second, "the sink-connector flow" constraints ensure that the flow through each of the "sink-connector" arcs is set equal to "p". Third, the "equi-distribution of flow" constraints ensure that the amount of flow through all arcs entering or leaving a facility node is equal. This last category of constraints requires four pairwise constraints at each facility node as shown.

Appendix C and D show that the solutions obtained from the LP83 formulation is integer and agrees with the MIP83 solutions. At issue, is whether or not this LP formulation would always provide an integer optimal solution given an integer right-hand-side and varying objective function coefficients. In other words, whether or not the constraint matrix is totally unimodular needs to be ascertained.

The proof that the constraint matrix is totally unimodular is beyond the scope of this research. The first step of this exercise would require Gaussian elimination to reduce the matrix to a (0,-1,1) matrix. If this can be accomplished, then algorithms are available to show whether or not the resulting matrix is totally unimodular (33:555). If the (0,-1,1) matrix is totally unimodular then the original constraint matrix is also totally unimodular (33:540).

1.5 Solution Procedure Using Branch and Bound and Microsolve Network Flow Programming. The integer solution to the problem can also be found by use of a Branch and Bound algorithm. The algorithm used to solve the maximal coverage problem is as follows (8):

Step 0. *Initialization.* Set Z_U to infinity (+).

Step 1. *Branch.* Select an unfathomed node and partition it into two subsets. (The subsets represent the two feasible values, 0 and 1, for flow through an arc leading out of the sink.)

Step 2. *Bound.* For each new subset, find a lower bound Z_L^i for the objective function value of feasible solutions in the subset, by solving a relaxed subproblem for the objective.

Step 3. *Fathom.* For each new subset i, exclude from further explicit enumeration (fathom) if:

(a) $Z_L^i \geq Z_U$;

(b) Subset i cannot have any feasible solution; or

(c) Subset i has a feasible solution. If $Z_L^i < Z_U$, set $Z_U = Z_L^i$ and store as the incumbent solution. Reapply test (a) to all unfathomed nodes.

Step 4. *Stopping Rule.* If no unfathomed nodes remain, stop. Incumbent solution is optimal. Else, return to Step 1.

The network parameters, as shown in Figure 2 and Tables 2, 3, and 4, were input to MICROSOLVE Network Flow Programming, and an initial solution to the relaxed problem (Z_L^o) was obtained. The output for the p=1 and p=2 problems are shown at Appendix E and F respectively. In these output, the source node is identified as node 19. A sink node was not explicitly provided. Rather, external flows at nodes 10 to 18 were set equal to -p. As can be seen from the output, the software created a slack node which is linked to the source and to nodes 10 to 18.

All relaxed subproblems were also solved using MICROSOLVE Network Flow Programming. Using this software, the flow through an arc can be set to a discrete value by setting both the lower bound parameter and the upper bound parameter for that arc equal to a specific value. Once the flow is fixed through all arcs requiring fixed flow for a given subproblem, the problem is solved again.

Since MICROSOLVE is interactive and menu-driven, the branch and bound algorithm is not too cumbersome for a problem of this size. Also, when 0-1 integer flow is imposed on all arcs emanating from the source, the flow through the remainder of the arcs in the network is automatically integer. This can be seen by inspection of Figure 2. Therefore, the stopping rule is invoked fairly quickly, with a maximum of $\{\Sigma\ 2^i\}$ subproblems to solve. With i=3, this only equates to a maximum of 14 subproblems, but, as 'i' increases, the number of subproblems that potentially need to be solved increases exponentially. However, one would expect that, in most if not all cases, a large proportion of the subproblems would not need to be solved because of the fathoming feature of the Branch and Bound algorithm.

The branch and bound solution trees for the p=1 and p=2 problems are shown at Figure 3 and Figure 4. The node numbers shown in the solution trees indicate the order in which the subproblems were solved. Both the p=1 and p=2 problems only required that 4 subproblems be solved. The optimal solution obtained in both cases agreed with the solutions found using MIP83 and LP83 above.

As shown in Figure 3, the initial solution to the relaxed p=1 problem was ZL=-69.996. This initial non-integer solution is a lower bound to the integer solution. The variables x1, x2, and x3, represent arcs S-1, S-2, and S-3 (as shown at Figure 2), respectively. The branching is first performed on the x1 variable. The x1=1 case

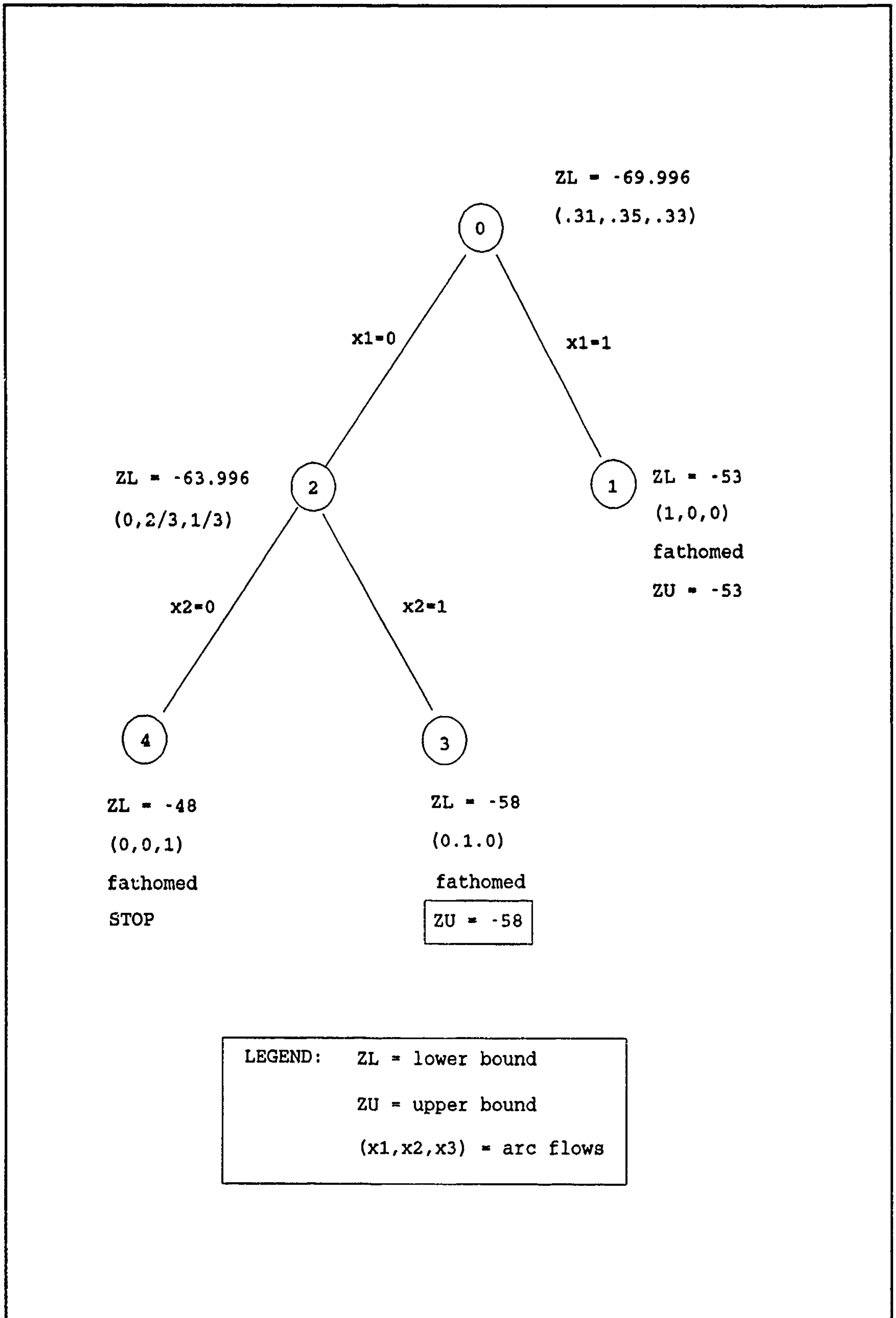


Figure 3 - Maximal Coverage Branch and Bound (p=1)

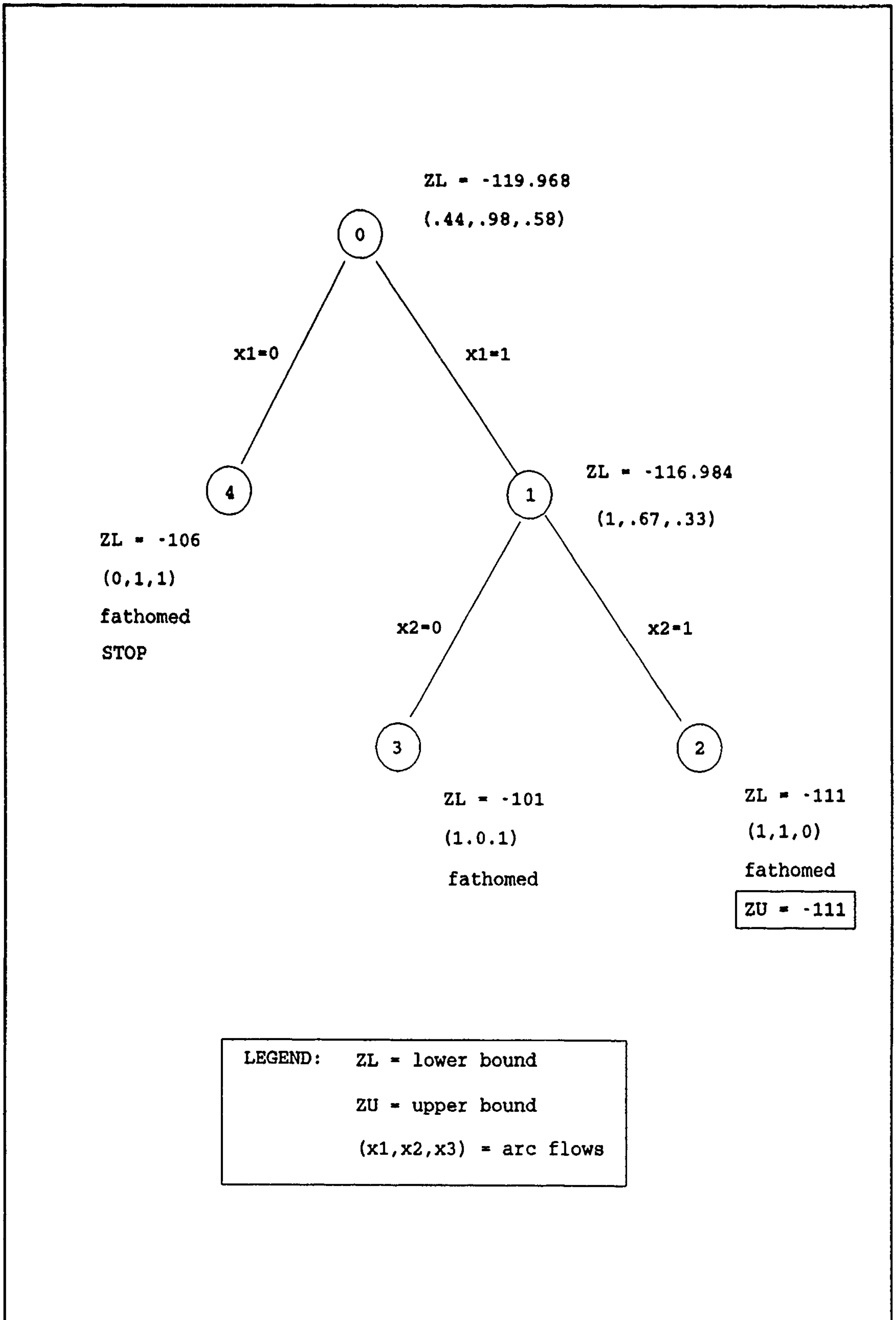


Figure 4 - Maximal Coverage Branch and Bound (p=2)

produces an integer solution and thus a new upper bound to the solution (ZU=-53). Node 1 is therefore fathomed. The x1=0 case does not produce an integer solution, and since the optimal solution to this subproblem (ZL=-63.996) is less than the current upper bound (ZU=-53) the node cannot be fathomed. Therefore node 2 is partitioned into two further subproblems. The node 3 subproblem produces a new upper bound (ZU=-58) which stands as the optimal solution once node 4 is fathomed because the node 4 subproblem produces a feasible, but sub-optimal solution.

The initial solution to the p=2 problem was ZL=-119.968. As mentioned above, the variables x1, x2, and x3 represent flow through arcs S-1, S-2, and S-3 of Figure 2. The node 1 subproblem (x1=1) was non-integer but produced a new lower bound (ZL=-116.984). The node 2 subproblem produced an integer solution and a new upper bound (ZU=-111). Nodes 3 and 4 also produced integer solutions and therefore these nodes were fathomed. But neither the node 3 solution nor the node 4 integer solution was less than the node 2 solution, and therefore the node 2 solution is optimal.

It is important to note that event though this is a maximal coverage problem, the subproblems are solved as min-cost problems in MICROSOLVE because the objective function coefficients are negative. Thus the initial solution to the relaxed problem produces a lower bound (the largest negative) on the optimal integer solution. Any integer solution will be greater (or less negative) than this relaxed solution (ignoring the case where the relaxed problem immediately provides an integer solution) and provide an upper bound on the optimal solution. The optimal solution is therefore the integer solution that produces the biggest negative value for the objective function. The absolute value of this optimal value represents the expected number of observation opportunities given the facilities selected.

Using network flow software instead of LP software to solve the subproblems improves the computational efficiency of the branch and bound algorithm. This advantage is derived from the fact that the basis can be represented by a tree in network algorithms. Therefore, "it is not necessary to store or computationally manipulate the basis inverse as in linear programming" (20:74).

1.6 Solution Procedure Using GAMS/BDMLP. The problem was also solved using GAMS (General Algebraic Modeling System). All results were duplicated and a typical output is shown at Appendix P and is described below.

The basic components of a GAMS model are (7:10):

SETS
  Declaration
  Assignment of members

Data (PARAMETERS,TABLES,SCALARS)
  Declaration
  Assignment of values

VARIABLES
  Declaration
  Assignment of type
  Assignment of bounds

EQUATIONS
  Declaration
  Definition

MODEL and SOLVE statements
DISPLAY statements

In the above list, the words that are fully capitalized indicate language keywords. These keywords serve to identify the nature of the statements that follow and appear before the next semi-colon.

As seen in Appendix P, the SETS group of statements are where the array subscripts are defined. Unlike FORTRAN 77 and many other high-level programming languages, the SET members do not take on integer values. For example, the subscript I which takes on the literal characters 1, 2, and 3, could be assigned location names such as Site1, Site2, and Site3.

The TABLE section is used to input the data associated with W_{ijk}. GAMS does not provide for the input of external data files. All data must appear in the source code at the beginning of the model as shown in the TABLE, PARAMETERS, and SCALARS section. Generally, data that is used in the model to produce coefficients or right-hand-side values are identified as PARAMETERS. A multi-dimensional parameter (of up to 10 dimensions) may also be declared and assigned data with the TABLE statement. The SCALAR statement is reserved for model parameters that have a dimensionality of zero (i.e., no associated SETS) and that are not variables in the problem.

The VARIABLES section is where the variable names and types are identified and also where the bounds on these variables can be specified. This is followed by the EQUATIONS section where the model equations are given names and defined algebraically. The MODEL statement allows for all or a part of the equations listed in the EQUATIONS section to be included in the model. This implies that more than one model can be solved in any given run but this was not attempted in this research.

The SOLVE statement directs which solver is to be used to solve the identified MODEL. Three solvers are available at AFIT which can solve the following types of problems:

BDMLP - linear programming

MINOS - linear and non-linear programming

ZOOM - zero-one and integer programming

Finally the DISPLAY statement determines which variable will be produced for output.

The main strength of the GAMS modelling software is obviously the ability to formulate the problem algebraically. This greatly reduces the potential for error as compared to software where all model equations need to be entered manually. GAMS provides excellent error handling to assist the user in "de-bugging" the model. This feature more than makes up for the less than adequate documentation provided with the software.

The inability to input data files is an inconvenience which can be overcome by merging ASCII files with the help of a word processor when building the GAMS model. However, for larger problems, this deficiency could become a serious nuisance.

2. *Set Covering Formulations*

As was discussed in Chapter II, the set-covering problem seeks to determine the minimum number of facilities such that all users are situated no more then the maximal desirable distance from the service location. Thus, unlike the p-median problem, the number of facilities, p, is a variable instead of a constant. The problem is formulated as follows:

$$\textit{minimize} \sum_{j \in I} x_{jj}$$

$$\textit{subject to}$$

$$x_{jj} - x_{ij} \geq 0 \qquad \forall\, i,j \in I \quad i \neq j$$

$$\sum_{j \in N_i} x_{ij} \geq 1 \qquad \forall\, i \in I$$

$$\textit{where}$$

$$N_i = \{j \mid d_{ij} \leq S\} \qquad \forall\, i \in I$$

The 0-1 integer variable, x_{jj}, is set to one when facility j is selected. The N_i variable ensures that, for any given demand location, the formulation only considers facilities that are less than the maximal distance. The constraints ensure that all locations are serviced and that the selected facilities are opened.

2.1 Mathematical Formulation. The above set-covering model requires only slight modifications to fulfill the needs of the GEODSS location problem. The objective function is identical, and seeks to minimize the number of facilities, p:

$$\textit{Minimize} \quad p$$
$$\textit{where } p \textit{ is an integer variable}$$

The observation opportunities for location *i*, in month *j*, of a type *k* satellite are computed as shown previously, but are labeled here as a_{ijk} instead of w_{ijk} in the maximal coverage formulation since observation opportunities are modeled as gains in the network flow formulation below. This change preserves the network flow programming notation used throughout this paper which shows gains with the letter 'a' and costs (or weights) with the letter 'w'. The minimum number of observation opportunities for a given mission 'k' is d_k, and is assumed to be constant in all states. This assumption is supported by operational requirement statements expressed by orbital analysts of Air Force Space Command (26).

The constraints are similar to the maximal coverage problem:

$$X_i - X_{ijk} \geq 0 \qquad \forall\, i,j,k$$

$$\sum_{i}^{I} X_i = p$$

$$\sum_{i}^{I} A_{ijk}\, X_{ijk} \geq d_k \qquad \forall\, j,k$$

$$\textit{where } X_{ijk} \textit{ is } 0\text{-}1 \textit{ integer}$$

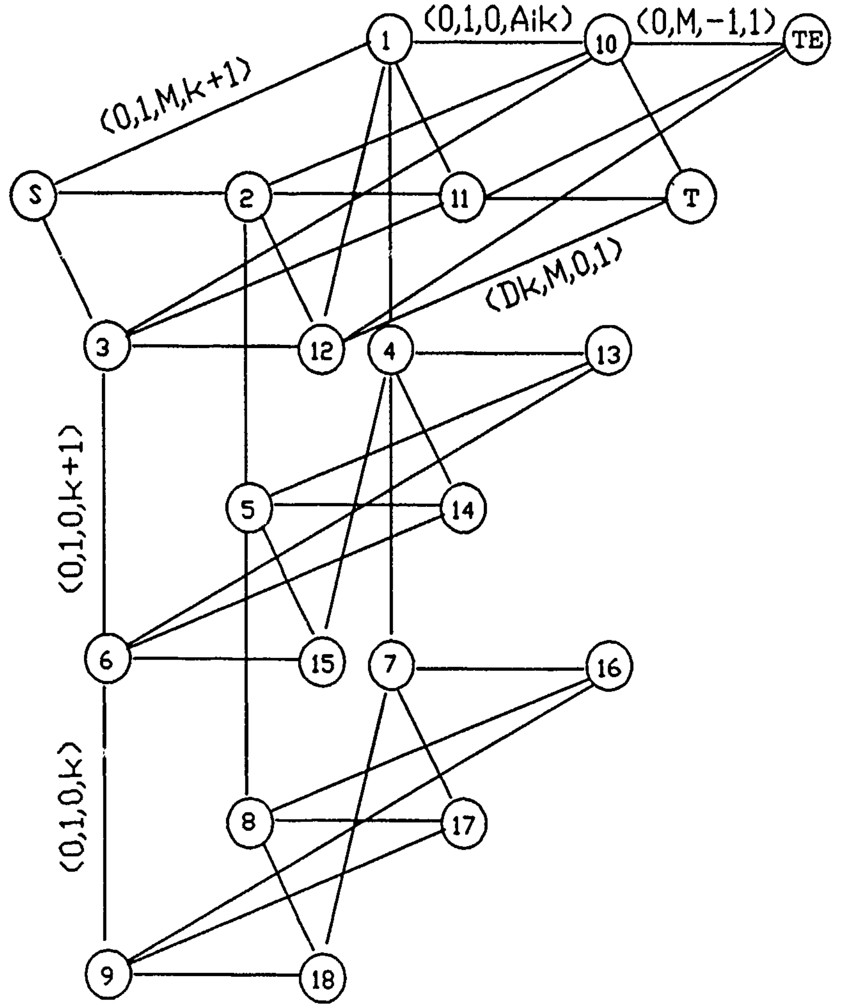
As with the maximal coverage formulation, the number of 0-1 integer variables is large enough to wa·rant an investigation of alternate solution methods.

2.2 Network-Flow Formulation. As with the maximal coverage problem, a transformation of the above mathematical formulation into a pure min-cost flow problem was not possible. A generalized min-cost network formulation was adopted.

The resulting network is shown in Figure 5. The source has a positive slack external flow (not shown). The arcs leading out of the source have an upper bound of 1 and a cost of 'M'. All other arcs have either a cost of zero, or, in the case of the arcs leading into the 'TE' node, a cost of -1. The gain of the arcs leading out of the source is (k+1) to allow for one unit of flow in all arcs leading out of a reached location. As with the maximal coverage network, nodes 1,4, and 7 represent location 1 in states 1,2, and 3 respectively. Similarly, nodes 2, 5, and 8 represent location 2 and nodes 3, 6, and 9 represent location 3. Also, nodes 10, 13, and 16 represent demand point 1, nodes 11, 14, and 17 represent demand point 2, and nodes 12, 15, and 18 represent demand point 3.

The arcs connecting locations and demand points have gains equal to the service capacity of that location-demand point pair. Therefore, since the flow through these arcs will either be 0 or 1 (as will be shown below), the flow into a given demand point represents the amount of service provided to that particular demand point. A lower bound for arcs leading from demand points to the 'T' sink ensures that the demand point requirement is met. The excess coverage is automatically routed to the arcs connecting the 'TE' sink since these have a cost set to -1. Therefore, the negative external flow at sink 'T' will equal ΣD_k, where k={10,...,18}. The external flow at sink 'TE' will equal $\Sigma\Sigma X_{ik}A_{ik} - \Sigma D_k$.

There is no requirement for the cost of all arcs leading into sink 'TE' to be set to -1. The relative costs of these arcs could for example represent relative merit of obtaining excess coverage for one satellite type as compared to another. This allows for a utility function to be constructed based on operational requirements. Perhaps the most practical way of assigning varying costs to the arcs leading into sink 'TE'



NOTES: (1) Arcs connecting nodes 13 to 18 to sinks T and TE are not shown

(2) (l,u,w,a) = lower bound, upper bound, weight, and gain

Figure 5 - Network Flow Diagram (Set Covering)

would be to make the assignments such that for each state 'j',

$$-1 < W_{k\text{-}TE} < 0 \quad \text{and} \quad \Sigma\, W_{k\text{-}TE} = -1.$$

The min-cost flow will minimize the slack external flow into the source, while encouraging excess coverage, or maximizing flow into the 'TE' sink. The minimum requirements at each demand node, which are expressed by the lower bounds on the arcs into the 'T' sink, determine how many arcs out of the source must be "activated".

Given 'i' locations, 'j' states, and 'k' demand points, the total number of nodes in the network is 3 (source and two sinks) plus the number of nodes in each state multiplied by the number of states: (3 + j(i+k)).

As was calculated for the maximal coverage problem network, the total number of arcs is the sum of, (1) the number of arcs out of the source, (2) the number of arcs into the sink, (3) the number of arcs connecting the facilities and demand points multiplied by the number of states, and (4) the number of "interstate" arcs. This reduces to: j(i+2k+ik).

Thus, the limited version of the problem, with i=j=k=3, includes 21 nodes and 54 arcs. The network representing the actual problem, which as stated previously should include i=10 locations, j=12 months, and k=10 satellite orbit classes, will therefore consist of 243 nodes and 1,560 arcs.

2.3 Solution Procedure Using MIP83. The above network formulation requires 0-1 integer flow through all arcs, except for the arcs into the sinks where integer flow is required. As mentioned previously, generalized network flow software does not guarantee integer flow since the network with gains constraint matrix is not totally unimodular.

As shown in the maximal coverage formulation, one way to force integer flow is of course to use integer programming software. MIP83 was used once again to obtain solutions to the problem.

The problem was first solved with the lower bound, D_k, on the arcs leading into sink 'T' set equal to 2. As shown in Appendix G, all variables were declared as 0-1 integer variables except for the variables representing arcs connecting the two sinks. The cost coefficients for arcs leading out of the source were set equal to 1000 so that these arcs would not enter the basis unless required to meet requirements specified by the D_k in the network. The "BOUNDS" section of the input is where the lower bounds D_k are identified. The "CONSTRAINTS" section includes m-1 nodal conservation of flow equations (where m is the number of nodes). The "NODE S" constraint is not a conservation of flow constraint but forces at least one facility to be selected. This constraint is redundant and could be omitted without affecting the solution in any way.

The optimal integer solution produced by MIP83 with $D_k = 2$ is to select facility 2 only. An inspection of Tables 2, 3, and 4, confirms the validity of this solution. As can be readily observed in these tables, all facilities provide a minimum of 2 observation opportunities to all satellite types. Therefore, from the point of view of meeting the minimum mission demands, all choices are feasible. However, facility 2 provides the largest number of observations overall (58) and therefore the largest overall excess coverage. Therefore, facility 2 surfaces as the optimal choice.

Note that the actual value of the objective function (960) provides an indirect measure of the amount of excess coverage provided by the selected facility or facilities. In this particular case, with one facility selected at a cost of 1,000, the amount of excess coverage is 1,000 - 960 = 40. This amount is a total of all 3 states. This can be again verified by inspection of Tables 2, 3, and 4. Noting that the sum of required coverage in each state is 6 (3 missions with requirements of 2 each), the excess provided by facility 2 is:

$$(18-6) + (18-6) + (22-6) = 40$$

which agrees with the above calculation.

Appendix H shows the output of the D_k=4 problem. As noted above, the requirements, D_k, appear in the "BOUNDS" section of the input. This is the only section of the D_k=2 problem that needed to be amended. MIP83 correctly selects facilities 1 and 2 given the lower bounds. This solution is again verified by inspection of Tables 2, 3, and 4 which reveal that facility 2 can no longer satisfy the increased requirement by itself. Therefore a minimum of two facilities are required. Sites 1 and 2 together meet the minimum demand and also combine to produce the largest total excess coverage since site 3 has the lowest total output over all states of 48 observations (14+15+19 = 48), as compared to site 1 which has a total of 53 observations (17+15+21 = 53), and site 2 which has a total of 58 observations (18+18+22 = 58).

Increasing the requirement to D_k=6 results in the selection of sites 1 and 3 (Appendix I). Table 3 shows that sites 1 and 2 together only provide 5 observations for mission 14 which falls short of the requirement. Therefore, the algorithm must either add site 3 to meet the requirement or select a different pair of sites. Since the objective seeks to minimize the number of facilities required to meet the demand, the algorithm finds that it can satisfy the demand by selecting sites 1 and 3. This choice is found to be optimal since it satisfies the primary objective of minimizing the number of facilities even though there is actually a decrease in excess coverage provided by the site 1 and 3 combination then was provided by sites 1 and 2 in the D_k=4 solution.

In summary, the formulation will select the minimum number of sites required to fulfill mission requirements. This is accomplished by assigning a large cost (M) to the selection of an arc leading out of the source. Discrimination between feasible solutions with equal number of sites is accomplished by rewarding total excess coverage. In this research, the amount of reward for excess coverage has been constant across mission types. However, a weighting scheme that would take into account varying mission priorities could be substituted.

2.4 Solution Procedure Using LP83. The D_k = 1, 2, and 3 problems were also solved using LP83. The LP83 output for these three problems are shown in Appendix J, K, and L, respectively.

The LP83 input was produced by removing the single and double square brackets of the corresponding MIP83 input. Nothing else in the MIP83 input file was modified. As was done in the maximal coverage LP83 formulation, "equi-distribution of flow" constraints were added in an attempt to force an integer solution on the problem.

The D_k=2 LP83 formulation produced the same integer solution as was produced by the MIP83 formulation. Site 2 was correctly selected has the optimal solution. However, the D_k=4 and the D_k=6 LP83 formulations did not return an integer solution. Analysis of the results show that this non-integer result is valid in the absence of an integer restriction on the flow out of the source. The model finds the minimum feasible flow out of the source that satisfies demands (as expressed by the lower bounds on the arcs into the sink 'T') and distributes the flow among the available arcs out of the source such that the equi-distribution of flow constraints are respected and a min-cost feasible optimal solution is produced. The only restriction on the flow out of the source is that it be greater or equal to 1 (NODE S constraint). In all three problems, the demands was satisfied with an external flow of 1 at the source. In the absence of the NODE S constraint, the problem will not return an integer solution for D_k=2 problem also.

Thus, even given the limited set of data used in this research, equi-distribution of flow constraints failed to produce integer solutions in all cases as was the case with the maximal coverage formulation. This result highlights the fact that the equi-distribution of flow constraints by themselves do not guarantee integer results.

Intuitively, these constraints must be coupled with fixed integer external flow at the source to produce an integer solution. However, total unimodularity of the constraint matrix needs to be established to show with complete certainty that a formulation will always return an integer solution.

2.5 Solution Procedure Using Branch and Bound and Microsolve Network Flow Programming. The Branch and Bound algorithm, as outlined in Section 2.4.3 of this report, was used to find the integer solutions to the D_k=2,4, and 6 set covering problems. MICROSOLVE Network Flow Programming was used once again to solve the relaxed initial and subproblems.

The initial relaxed solution to the D_k=2 problem is shown in Appendix M. This initial non-integer solution placed a lower bound of 258.267 on the integer solution. As shown in Figure 6, the node 1 subproblem was feasible and an upper bound of 965 was calculated by MICROSOLVE. Node 3 returned a new upper bound of 960 which became the optimal solution once the remainder of the nodes were fathomed. This solution agrees with the MIP83 solution to the same problem.

The solution to the initial relaxed D_k=4 problem can be found at Appendix N and the branch and bound solution tree is shown at Figure 7. The algorithm again produced the same solution as the MIP83 solution. The D_k=6 problem was also solved using the branch and bound algorithm and produced the same solution as MIP83. The D_k=6 initial relaxed problem solution is at Appendix O and the branch and bound solution tree is shown at Figure 8.

Not including the initial relaxed problem, six subproblems had to be solved for the D_k=2 problem, 10 for the D_k=4 problem, and 8 for the D_k=6 problem. Therefore, on the average, eight subproblems needed to be solved which is about twice the number of subproblem solutions that was required in the maximal coverage formulation. This greater number of branching in the set covering formulation may be due to the fact that the source external flow is slack instead of fixed. More

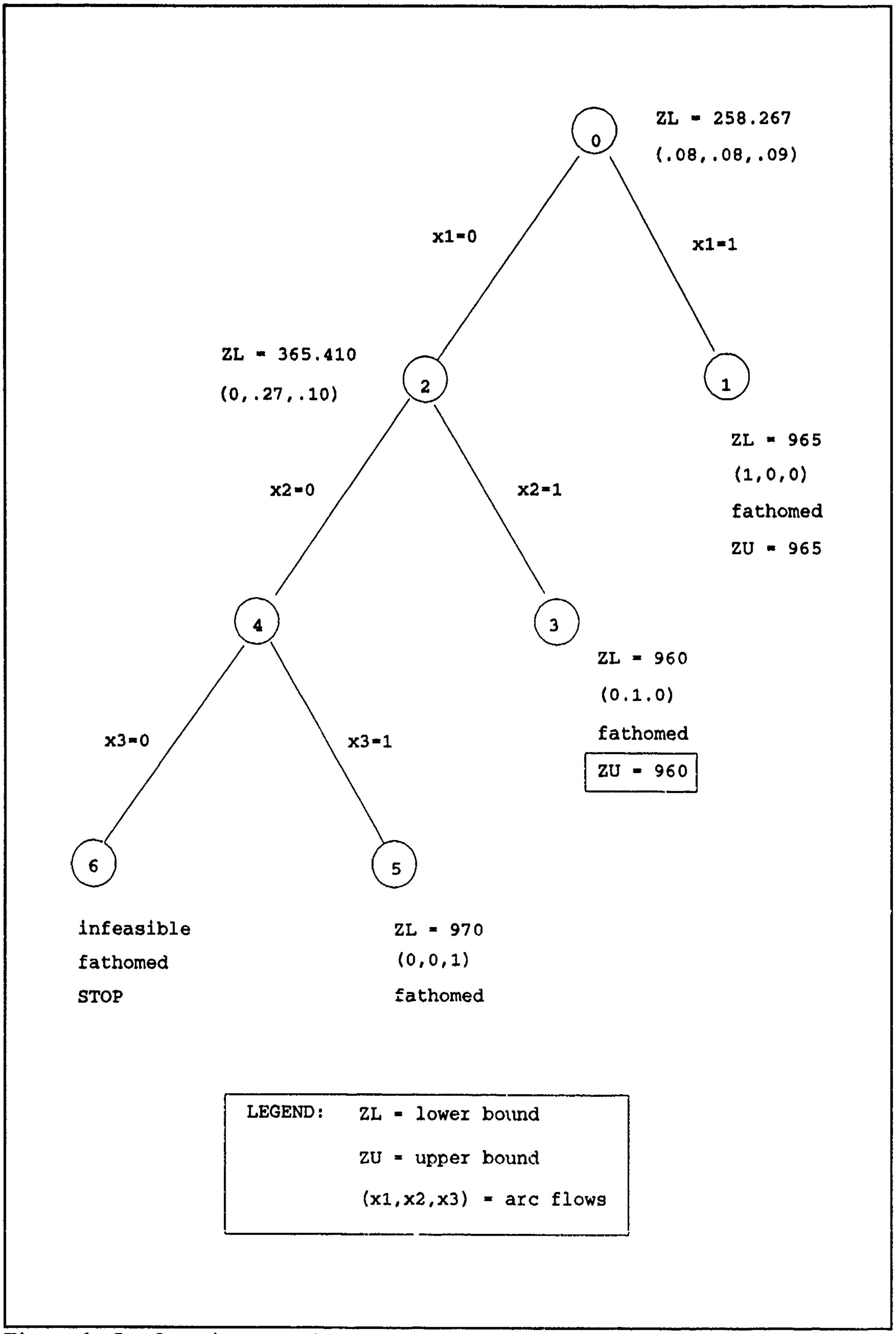


Figure 6 - Set Covering Branch and Bound (D_k=2)

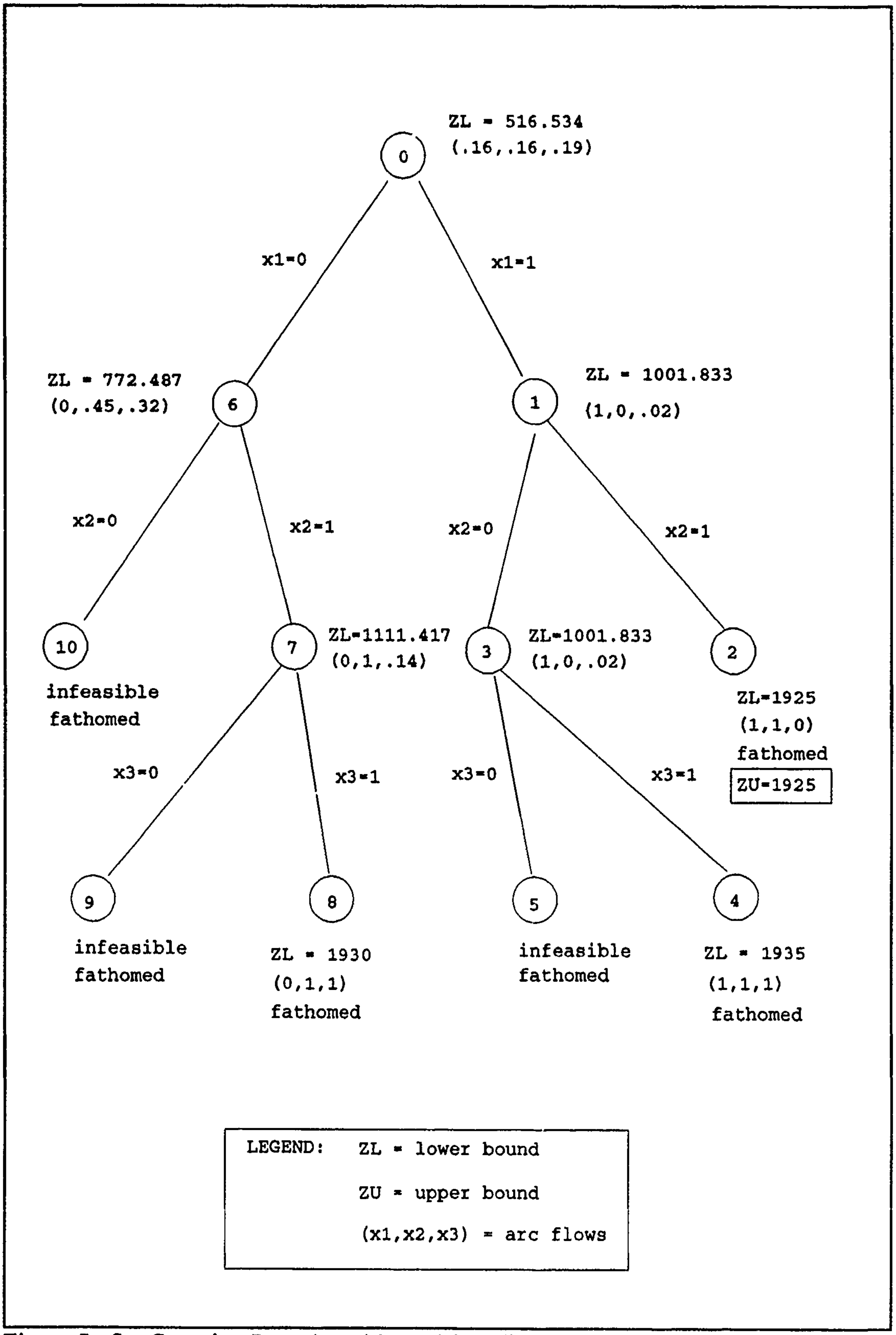


Figure 7 - Set Covering Branch and Bound (D_k=4)

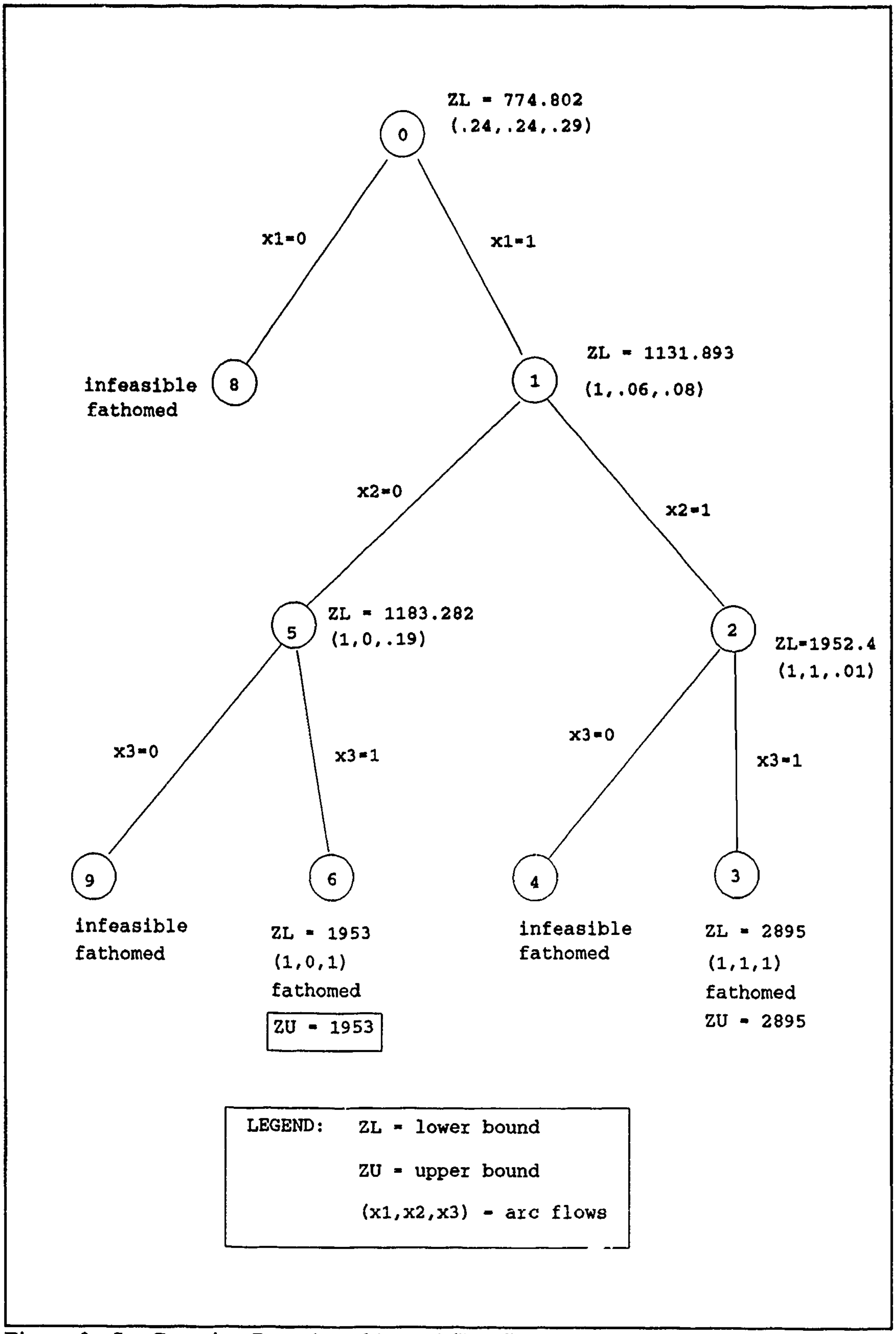


Figure 8 - Set Covering Branch and Bound (D_k=6)

experimentation with varying sets of data would be required to establish a correlation. As with the maximal coverage branch and bound formulation, the maximum number of subproblems that need to be solved to complete the algorithm is $\Sigma\ 2^i$ where 'i' is the number of locations. For i=3, the maximum is 14.

2.6 Solution Procedure Using GAMS/BDMLP. The problem was also solved using GAMS (General Algebraic Modeling System). All results were duplicated and a typical output is shown at Appendix Q (GAMS is described in Section 1.6).

3. Conclusions

By definition, both the maximal coverage and set covering problems are 0-1 integer problems. These types of problems are NP complete and, depending on the number of variables involved, may be difficult or impossible to solve using MIP software because the number of feasible solutions that must be evaluated is explosive.

The network flow formulations in this research have resulted in problems with approximately the same number of arcs as there were variables in the mathematical formulations of the maximal coverage and set covering problems. Thus, the graphical representation of these problems has not reduced the size of the problems. However, the network flow model does provide an increased understanding of the dynamics of the problem by allowing the user to visualize the progress of units of flow through the network and therefore attach physical meaning to a mathematical solution process.

One major advantage of the network model was to allow for the use of network flow programming software to solve the subproblems in the Branch and Bound algorithm. Since the network flow programming subproblems are not NP complete, computational efficiency is better than solving the subproblems as standard linear programs which are NP complete.

A software package that interfaces a Branch and Bound algorithm and generalized network flow programming software would be ideal for solving the full size problem ($i=10$, $j=12$, $k=10$) which includes approximately 250 nodes and 1500 arcs. Software of this kind is not readily available. A manual solution using the Branch and Bound algorithm is not plausible because of the large number of subproblems that potentially need to be solved in the full scale problem. Intuitively, once the arcs leading out of the source in both models have been integerized, the remainder of the arcs in the network must also carry integer flow. While this characteristic of the network would greatly reduce the amount of branching required by the algorithm, the number of subproblems requiring solution could still reach $\Sigma\ 2^i$. With $i=10$ this amounts to 2,046 subproblems which clearly is prohibitive for manual computations.

A potential major advantage of the network flow formulation is that it may be providing a way of significantly reducing the number of 0-1 integer variables in both the maximal coverage and set covering problems. If imposing 0-1 integer restriction on the arcs leading out of the source does indeed force integer flow in the remainder of the network, then the total number of 0-1 variables would be equal to the number of locations 'i', and not the product 'ijk' of the original mathematical formulation. This means that the network flow formulation decreases the number of 0-1 integer variables in the problem by a factor of $1/jk$ which would open up the option of solving the GEODSS location problem using MIP software.

Further investigation in solution procedures for this problem should be motivated by ways of improving computational efficiency. For example, polyhedral description techniques, such as the Gomory cutting-plane (33:367) and Lagrange relaxation (33:323) algorithms and/or a combination of both of these, might be attempted.

V. MULTIOBJECTIVE MODELS

1. Maximal Coverage Formulation

The network formulation presented in Chapter IV provided a good framework for addressing the objective of maximizing the number of observation opportunities, w_{ijk}. A second objective function, that of minimizing the variance in the number of observation opportunities per satellite needs to be defined.

1.1 Variance Criterion Function. The mean number of observation opportunities provided to a satellite in month j is given by:

$$U_{ij} = \sum_{k}^{K} W_{ijk} \times \frac{1}{K}$$

Therefore, for a given location i, the variance in the number of observation opportunities provided to a satellite in month j is given by:

$$V_{ij} = \sum_{k}^{K} (W_{ijk} - U_{ij})^2 \times \frac{1}{k}$$

Using the data in Tables 2, 3, and 4, the mean and variances in the number of observation opportunities can be calculated. The results are shown at Tables 5 and 4.

The objective function to minimize the sum of the variances is given by:

$$Minimize\ z = \sum_{i}^{I} \sum_{j}^{J} V_{ij}\, X_i$$

1.2 Problem Formulation. The multicriteria optimization problem is formulated by adding the variance objective function to a linear programming formulation of the network at Figure 2. As indicated in Chapter IV, the node-arc incidence matrix of the network is not totally unimodular and an integer solution is not guaranteed. However, imposing a binary integer restriction only on the variables for the arcs leading out of the source is sufficient when a series of side constraints that impose conditions on the flow through the arcs of the network are added to the node-arc incidence matrix. As shown at Appendix R, these are categorized as "sink-connector flow" or "equi-distribution of flow" constraints.

1.3 Correlation of Objectives. The degree to which objectives are correlated must be determined. If sufficiently strong correlation can be shown, then the multi-objective problem can be solved as a single-objective problem by optimizing one or the other objective functions. Furthermore, if strong correlation exists, solving the multi-objective problem with the use of a weight vector may generate inconsistent results (39:198).

A measure that suffices for the degree to which the i^{th} and j^{th} objectives are correlated is the angle between the gradients c^i and c^j, which can be calculated as follows (39:198):

$$\alpha = \cos^{-1} \left(\frac{(c^i)^T c^j}{\|c^i\|_2 \ \|c^j\|_2} \right)$$

The smaller the angle, the more the gradient vectors take on the same orientation and therefore, the more correlated the objectives. For the multicriteria maximal coverage formulation, the angle between the gradient vectors takes on the maximum value of 90 degrees. This can be seen by inspection of the two equality constraints "OBJ1" and "OBJ2" (Appendix R) which represent the two objective functions of the problem. Only the decision variables with non-zero coefficients are shown in these two constraints. Since all variables in these objectives only appear in

i/j	1	2	3
1	5.667	5	7
2	6	6	7.333
3	4.667	5	6.333

Table 5 - U_{ij}

one or the other objective, the dot product in the above formula, $(c^1)^T c^2$, is zero and therefore the angle between the vectors is 90 degrees. This indicates zero correlation between the two objectives of the multicriteria maximal coverage formulation.

1.4 Generation of the N-Set. Two important facts concerning this problem must be noted. First, since the number of observation opportunities is maximized and the variance is minimized, then the condition that "more is better" exists for both objective functions. Thus, Pareto preference is assumed and outcome y^1 is preferred to outcome y^2 iff $y^1 \geq y^2$ (50:10). Second, given that this is an integer programming problem, the alternative space X is not necessarily convex. Given a non-integer programming problem and Pareto preference, the entire set of non-dominated points (N-points) with respect to Pareto preference can be found by using "appropriate mathematical programming techniques" and by varying the weights associated with

i/j	1	2	3	TOTAL
1	2.889	4.667	4.667	12.223
2	6	6	1.556	13.556
3	4.222	0.667	6.222	11.111

Table 6 - V_{ij}

each objective function over the preferred cone (50:33). As will be shown below, this procedure, given the name "weighted-sum method" in this research, does not guarantee the complete N-set will be generated in the case of integer programming problems.

Another technique of generating the set of N-points is to transform one of the objective functions f_i into a constraint and then vary the right hand side of this constraint from the lower bound to the upper bound of f_i over the alternative space, X (50:33). Unfortunately, this technique also does not guarantee a complete N-set will be generated for an integer problem.

If the combinatorial problem is not exceedingly large, one approach is to generate an exhaustive list of the points in the alternative space and map each of these points to the outcome space. The N-set can then be defined with respect to Pareto preference. This technique does guarantee that the entire N-set will be generated, but is of course only applicable to problems with a relatively small number of feasible alternatives.

1.5 Weighted-Sum Method.

1.5.1 Formulation.

The MIP83 program shown at Appendix R was executed for various combinations of objective function weights. The two objective functions are represented by the F1 and F2 variables as shown at the beginning of the ..OBJECTIVE section of the program. These two variables take on values according to the two equality constraints, OBJ1 and OBJ2, shown in the ..CONSTRAINTS section of the program. OBJ1 represents the maximization of observation opportunities and OBJ2 the minimization of variance. Accordingly, the coefficients of the variables in these two constraints are taken from Tables 2, 3, and 4 for OBJ1, and from Table 6 for OBJ2.

The node-arc incidence matrix constraints (conservation of flow) are labeled according to the node numbering scheme of Figure 2. As mentioned above, the "sink-connector" and "equi-distribution of flow" constraints ensure integer flow through the network given that the arcs out of the source have been integerized. The "select p facilities" equality constraint is set to the number of facilities to be selected.

1.5.2 Results and Analysis. For the p=1 problem, the alternative space consists of choosing between location 1, 2, or 3. By inspection of Tables 2, 3, 4 and 6, the outcomes associated with these alternatives are:

$$y^1 = (53, -12.223)$$

$$y^2 = (58, -13.556)$$

$$y^3 = (48, -11.111)$$

The MIP83 program was executed for various combinations of weights for the objective functions. The results are as shown in Table 7.

λ1	λ2	X^* (p=1)	X^* (p=2)
0	1	Site 3	Sites 1,3
.1	.9	Site 3	Sites 1,3
.2	.8	Site 2	Sites 1,2
.3	.7	Site 1	Sites 1,2
.4	.6	Site 1	Sites 1,2
.5	.5	Site 1	Sites 1,2
.6	.4	Site 1	Sites 1,2
.7	.3	Site 1	Sites 1,2
.8	.2	Site 1	Sites 1,2
.9	.1	Site 1	Sites 1,2
1	0	Site 1	Sites 1,2

Table 7 - Multicriteria Max Coverage Optimal Solutions

A plot of the outcome space (Figure 9) validates the results shown in Table 7. From this plot, the two watershed points were found to be $(\lambda 1,\lambda 2) = (.1819,.8181)$ where there is indifference between sites 1 and 3, and (.2159,.7841) where there is indifference between sites 1 or 2. The parametric decomposition is shown in Figure 10.

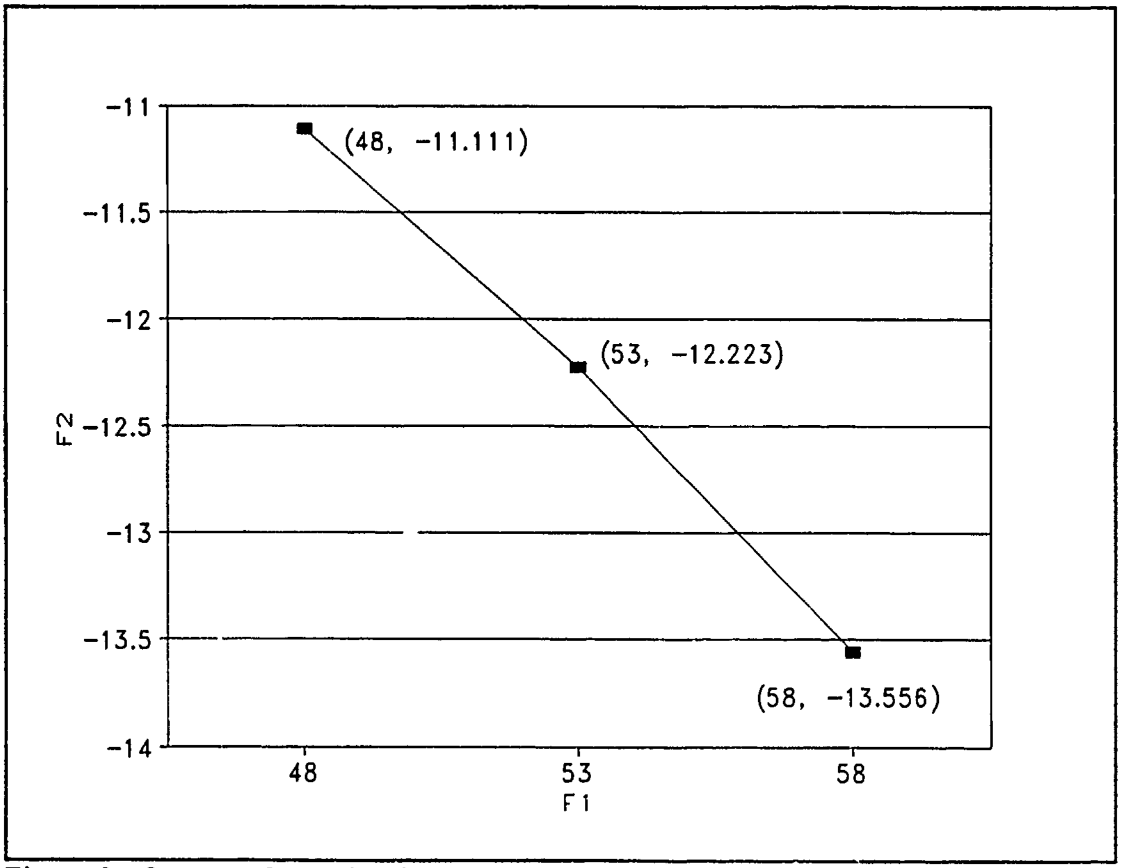


Figure 9 - Outcome Space (p=1)

As can be seen from inspection of Figure 10, a line joining the three points of the outcome space for the p=1 problem forms a convex surface. Consequently, the complete set of N-points, which in this case only consists of y^1, y^2, and y^3, was obtained by varying the weights of the objective functions.

The p=2 maximal coverage problem alternative space also consists of only three points. The optimal solution must be chosen from one of the following three pairs of locations: sites 1 and 3, sites 2 and 3, and sites 1 and 2. By inspection of

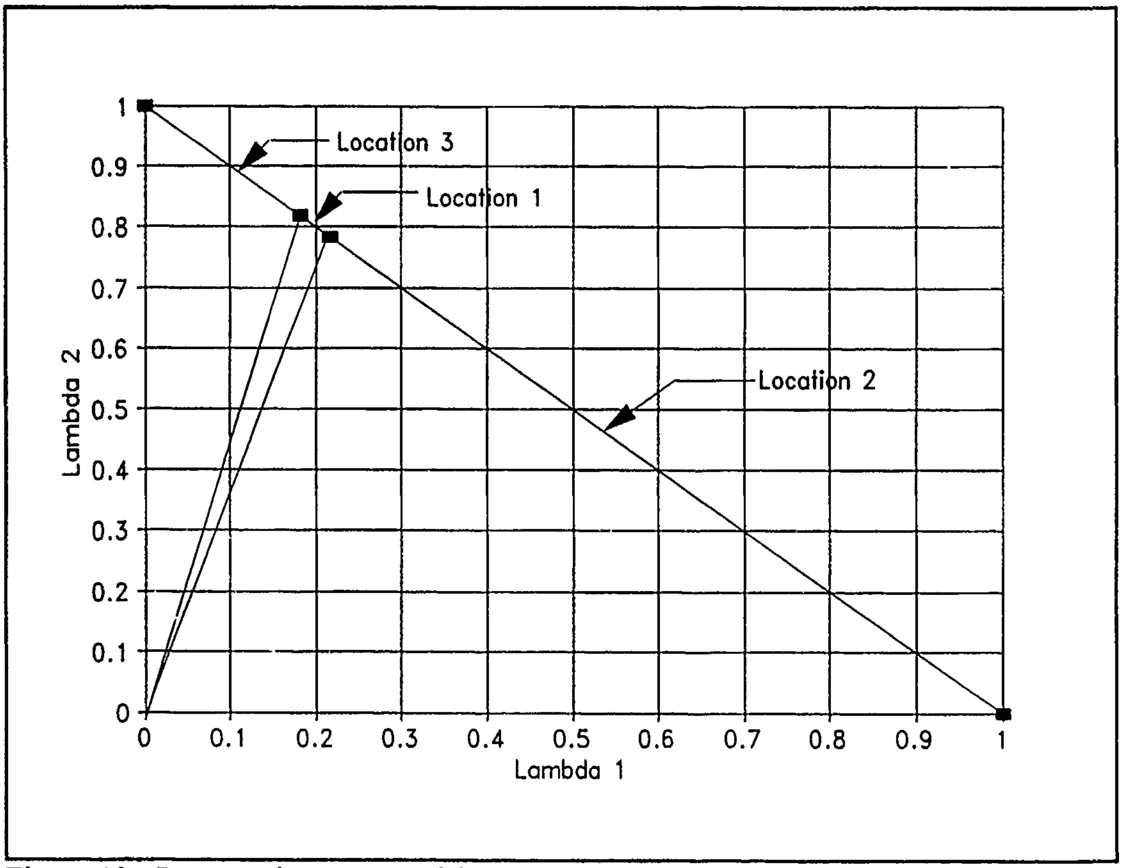


Figure 10 - Parametric Decomposition (p=1)

Tables 2, 3, 4, and 6 the outcomes associated with these three alternatives are:

Sites 1,2 $y^1 = (101, -23.334)$

Sites 2,3 $y^2 = (106, -24.667)$

Sites 1,2 $y^3 = (111, -25.779)$

Unlike the p=1 problem, a plot of the outcome space shows that a line joining the three N-points does not form a convex space. As shown in Figure 11, y^2 is actually to the left of a line drawn from y^1 to y^3. Consequently, as the assumption of convexity discussed above is not valid for the p=2 problem, varying the weights of the objective function did not reveal the y^2 N-point. As shown in Table 7, the column of optimal alternatives does not include the y^2 N-point since the value function $z = \lambda_1 f_1 + \lambda_2 f_2$ never touches this point as the slope $(-\lambda_1/\lambda_2)$ of the function is varied from zero to infinity.

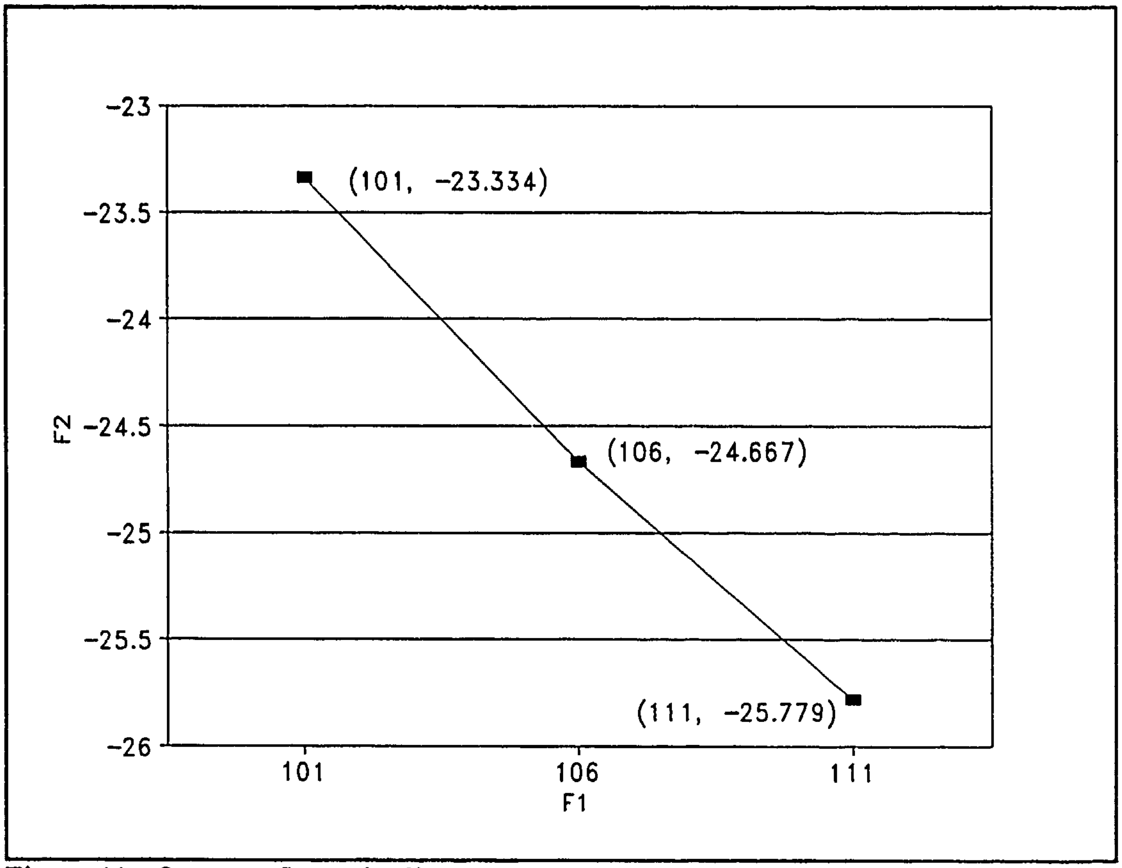


Figure 11 - Outcome Space (p=2)

Given that the y^2 was not discovered by the weighted sum approach, the parametric decomposition shows that the watershed point between the choice for Sites 1 and 3 and Sites 1 and 2 occurs at $(\lambda_1,\lambda_2) = (.1965, .8035)$. The parametric decomposition is shown in Figure 12.

The fact that the weighted sum approach failed to reveal the existence of one of the N-points underscores the problem in multicriteria optimization of problems with integer restrictions. Since the lack of convexity means that not all N-points will be generated, viable solutions to the problem may be overlooked. In the p=2 problem, the three outcomes nearly form a straight line in the outcome space. This means that, in actuality, the watershed point nearly includes all three alternatives. In a larger problem where it is not possible to list all outcomes, this fact would not have surfaced.

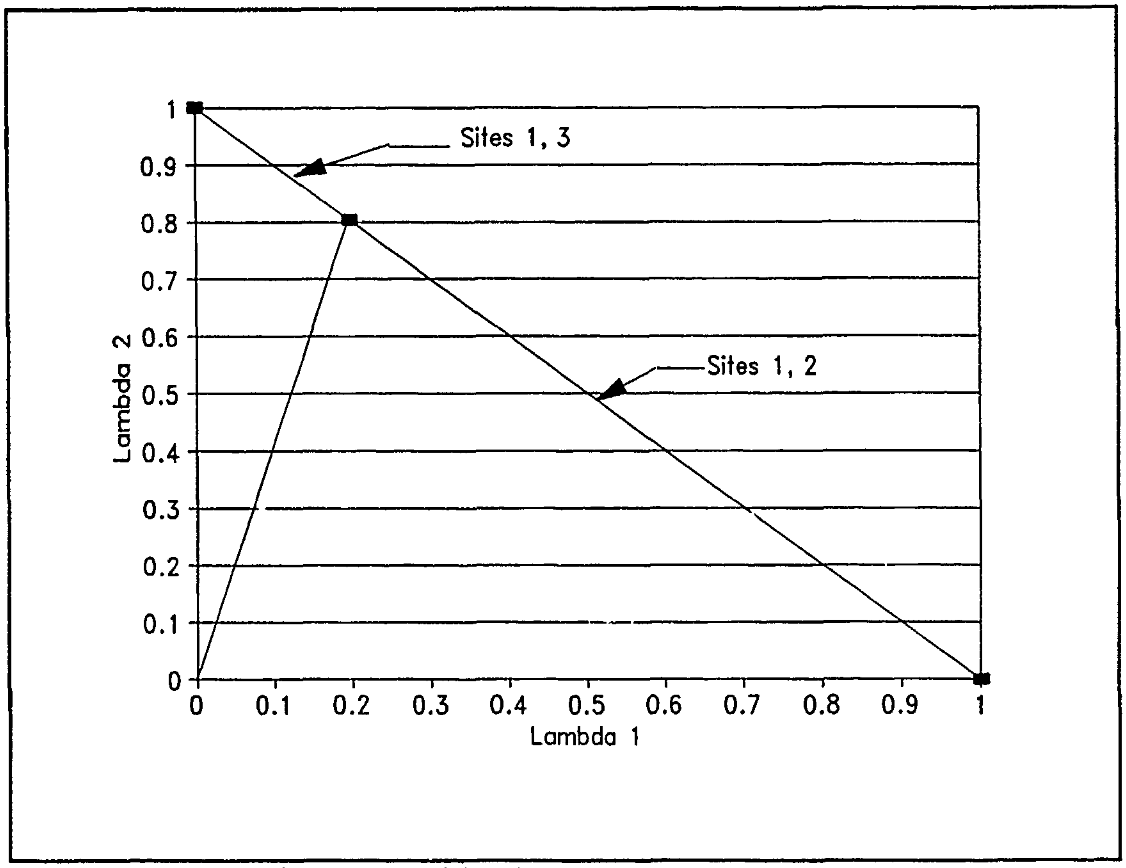


Figure 12 - Parametric Decomposition (p=2)

1.6 Constraint Method.

1.6.1 Formulation. The MIP83 weighted-sum method formulation was modified to produce the constraint method formulation. As shown in Appendix S, only the variable F1 (representing the network flow objective) is optimized, and variable F2 (representing the variance criterion function) is constrained by a new constraint labeled "SATISFICE". As indicated previously, the efficient frontier is generated by repeatedly solving the F1 optimization probler /ith the "SATISFICE" constraints ranging from the lower to the upper bound of F2. These bounds are found by "throwing out" F1 and minimizing and maximizing F2 over the alternative space.

An equality constraint, which appears immediately following the network nodal conservation of flow constraints, is used to set the number of facilities, p, to be selected.

1.6.2 Results and Analysis. With p=1, the lower and upper bounds of F2 were found to be 11.110 and 13.556, respectively. This result is validated by inspection of Table 4. Given the difference between the lower and upper bounds of F2 is 2.445, it was arbitrarily decided to vary the right-hand-side of the "SATISFICE" constraint in increments of 2.445/10 = .2445 to generate the efficient frontier. The results are shown in Table 8.

Referring back to the plot of the outcome space for the p=1 problem (Figure 9) we see that the constraint method was successful in generating the three N-points for this integer problem.

SATISFICE LEVEL	F1	X^*
13.5560	58	Site 2
13.3115	53	Site 1
13.0670	53	Site 1
12.8225	53	Site 1
12.5780	53	Site 1
12.3335	53	Site 1
12.0890	48	Site 3
11.8445	48	Site 3
11.6000	48	Site 3
11.3555	48	Site 3
11.1110	48	Site 3

Table 8 - Constraint Method Efficient Frontier (p=1)

With p=2, the lower and upper bounds of F2 were found to be 23.334 and 25.779, respectively. This result is also validated by inspection of Table 6. As with the p=1 problem, the difference between the lower and upper bounds of F2 is 2.445, and the right-hand-side of the "SATISFICE" constraint is also varied in increments of 2.445/10 = .2445 to generate the efficient frontier. The results are shown in Table 9.

Unlike the weighted-sum method which had not revealed the "Site 2-3" outcome, the constraint method is found to successfully generate all three N-points of the outcome space (Figure 11).

Obviously, whether or not an N-point is missed using the constraint method is a function of the size of the incremental steps taken in varying the right-hand-side

SATISFICE LEVEL	F1	X^*
25.7790	111	Sites 1,2
25.5345	106	Sites 2,3
25.2900	106	Sites 2,3
25.0455	106	Sites 2,3
24.8010	106	Sites 2,3
24.5565	101	Sites 1,3
24.3120	101	Sites 1,3
24.0675	101	Sites 1,3
23.8230	101	Sites 1,3
23.5785	101	Sites 1,3
23.3340	101	Sites 1,3

Table 9 - Constraint Method Efficient Frontier (p=2)

of the SATISFICE constraint. Therefore, this method does not guarantee that the entire N-set will be generated.

1.7 Exhaustive List Algorithm. For the combinatorial problem of selecting "p" facilities from a set of "i" locations, the total number of possible solutions is given by $i!/(i-p)!p!$. Thus, while the number of solutions is explosive with large "i", it is possible to enumerate all feasible solutions for problems with a limited number of candidate locations and a given "p".

The following statement follows from the definition of an N-point (50:15). Given two criteria functions, f_1 and f_2, a feasible integer solution, x^*, is an N-point, y^*, in the outcome space iff for all other points, y^j, in the outcome space the following logical statement is false (i.e., at least one of the inequalities is false):

$$(y_1^j \geq y_1^*) \text{ AND } (y_2^j \geq y_2^*)$$

If both of the above inequalities are true, then the logical statement is true and y^* is a D-point (dominated point).

The above discussion can be summarized in an algorithm to generate the efficient frontier for relatively small combinatorial problems:

Step 1. Generate the alternative space by testing each possible combinations of the decision variables against the constraints (if any).

Step 2. Map each point in the alternative space to the outcome space using the criteria functions.

Step 3. Determine set of N-points by testing each point in the outcome space using the above condition.

Notwithstanding the limitation in problem size, when compared to the weighted-sum and constraint methods, the above algorithm has the advantage of ensuring that all N-points will be generated. However, in combinatorial problems that include non-binary integer variables, problem size becomes even more restrictive since each level of the integer variable will potentially produce a new set of feasible solutions.

1.8 Compromise Solutions. With compromise programming, the optimal alternative is the alternative that minimizes the norm, r, for given values of λ and p. The norm is given by

$$r(y;p,\lambda) = \Big[\sum_i \lambda_i^p |y_i - y_i^*|^p \Big]^{\frac{1}{p}}$$

For the purposes of this research, we assume the weights are equal and investigate the impact of varying the parameter p on the ranking of the alternatives. The results for the p=1 problem are shown in Table 10.

The results show that increasing p does not change the ranking of the alternatives. This is explained by the fact that we are dealing with discrete points in the outcome space. Varying p has varied the shape of the weighted metric, but not

X^*	$r(y;1)$	$r(y;2)$	$r(y;\infty)$
Site 2	2.445	2.445	2.445
Site 1	6.112	5.122	5
Site 3	10.000	10.000	10.000

Table 10 - Norms (p=1)

sufficiently enough to disrupt the ranking of alternatives by picking up an adjacent N-point as the optimal choice. Intuitively, whether or not varying p will produce varying optimal choices for a given set of weights is a factor of the number and distribution of discrete N-points in the outcome space. More experimentation is needed to definitively show this.

Similar results were obtained for the p=2 problem and these are shown in Table 11.

X^*	r(y;1)	r(y;2)	r(y;∞)
Sites 1,2	2.445	2.445	2.445
Sites 2,3	6.330	5.175	5
Sites 1,3	10.000	10.000	10.000

Table 11 - Norms (p=2)

2. *Set Covering Formulation*

2.1 Variance Criterion Function. The variance criterion function for the set-covering formulation is identical to the variance criterion function of the maximal coverage problem. It is repeated here for convenience:

$$\textit{Minimize } z = \sum_{i}^{I} \sum_{j}^{J} V_{ij} X_i$$

The coefficients, V_{ij}, represent the variance at location i, in month j, and are tabulated in Table 6.

2.2 Problem Formulation. The multicriteria set-covering problem is formulated by adding the variance objective function to a linear programming formulation of the network at Figure 5. As with the maximal coverage formulation, the node-arc incidence matrix of the network is not totally unimodular and an integer solution is not guaranteed. However, imposing a binary integer restriction on the variables for the arcs leading out of the source is sufficient to guarantee an integer solution when a series of side constraints that impose conditions on the flow through the arcs of the network are added to the node-arc incidence matrix. As shown in Appendix T, these are categorized as "sink-connector flow" or "equi-distribution of flow" constraints.

2.3 Correlation of Objectives. The degree to which the objectives are correlated is determined by calculating the angle between the gradient vectors using the formula presented earlier:

$$\alpha = \cos^{-1}\left(\frac{(c^i)^T c^j}{|c^i|_2\ |c^j|_2}\right)$$

Extracting the objective function coefficients from the OBJ1 and OBJ2 constraints in Appendix T, the angle between the two gradient vectors is found to be approximately 4.6 degrees. This would seem to indicate a high degree of correlation between the two objective functions and allow solution to the multicriteria optimization problem by solving one or the other single objective problem.

Nonetheless, the exercise of generating the efficient frontier is undertaken to attempt to identify any possible difficulties caused by the high degree of correlation.

2.4 Generation of the N-set. Given that the research with the maximal coverage problem has already revealed the deficiency in using the weighted-sum method to generate the efficient frontier, this method will not be attempted here. The N-set will be generated using the constraint method and an exhaustive-list algorithm, similar to the one developed in the maximal coverage problem, will be proposed.

2 5 Constraint Method.

2.5.1 Formulation. The MIP83 program shown in Appendix T was executed for various combinations of objective function weights. The two objective functions are again represented by the F1 and F2 variables as shown in the beginning and end of the ..OBJECTIVE section of the program. These two variables take on values according to the two equality constraints, OBJ1 and OBJ2, shown in the

..CONSTRAINTS section of the program. OBJ1 represents the minimization of cost for the network, and the coefficients for the variables are as described above. OBJ2 represents the minimization of variance. Accordingly, the coefficients for the OBJ2 variables are taken from Table 6. The "SATISFICE" constraint is used to vary the level of the F2 objective.

The node-arc incidence matrix constraints (conservation of flow) are labeled according to the node numbering scheme of Figure 5. As mentioned above, the "sink-connector" and "equi-distribution of flow" constraints ensure integer flow through the network given that the arcs out of the source have been integerized. The ..BOUNDS section, which did not appear in the maximal coverage formulation, is where the minimum demand point requirements are specified.

2.5.2 Results and Analysis. Three distinct set covering problems were attempted by setting the demand point requirements, D_k, equal to 2, 4, and 6, respectively, for all "k". For each of these three problems, the range of F2 was determined as before by minimizing and maximizing F2 over the feasible region.

With D_k=2, the range of F2 is found to be 11.111 to 36.890. Using the constraint method, the complete set of N-points is generated as shown in Table 12. All results are consistent with the data as shown in Tables 2 to 4. The outcome space is shown in Figure 13.

With D_k=4, F2 ranges from 23.334 to 36.890. The complete set of N-points is generated using the constraint method. The results are shown in Table 13 and are also consistent with the data shown in Chapter IV. The outcome space is shown in Figure 14.

With D_k=6, it can be seen by inspection of Tables 2, 3, 4 of Chapter IV that there are only two feasible solutions that meet this level of demand: selecting Sites 1 and 3 or selecting all three sites. Therefore, as the 3-site alternative is obviously dominated, the outcome space consists of only one N-point. Using the constraint

SATISFICE LEVEL	F1	X^*
36.8900	960	Site 2
13.5560	960	Site 2
13.0670	965	Site 1
12.5780	965	Site 1
12.0890	970	Site 2
11.6000	970	Site 3
11.1110	970	Site 3

Table 12 - Multicriteria Set Covering Solutions (d=2)

method with F2 ranging from 23.334 to 36.890 confirms this result.

The high degree of correlation between the two objective functions does not appear, at least on the basis of this research, to produce erratic results. In fact, in this problem, the ranking of alternatives obtained in optimizing one objective is opposite that obtained in optimizing the other. For example, with D_k=2, the min cost ranking of alternatives would be Site 2, followed by Site 1 and Site 3. Conversely, the min variance ranking would be Site 3, followed by Site 1 and Site 2. Therefore, even though the angle between the gradient vectors is small, it does not appear that optimizing one objective will implicitly optimize the other.

2.6 Exhaustive List Algorithm. The 3-step algorithm developed for the maximal coverage problem is modified slightly for the set-covering problem. The mapping from the discrete alternative space into the discrete outcome space is slightly complicated by the fact that the level of the variables in the F1 objective representing the arcs into the excess sink must be determined for each combination of sites:

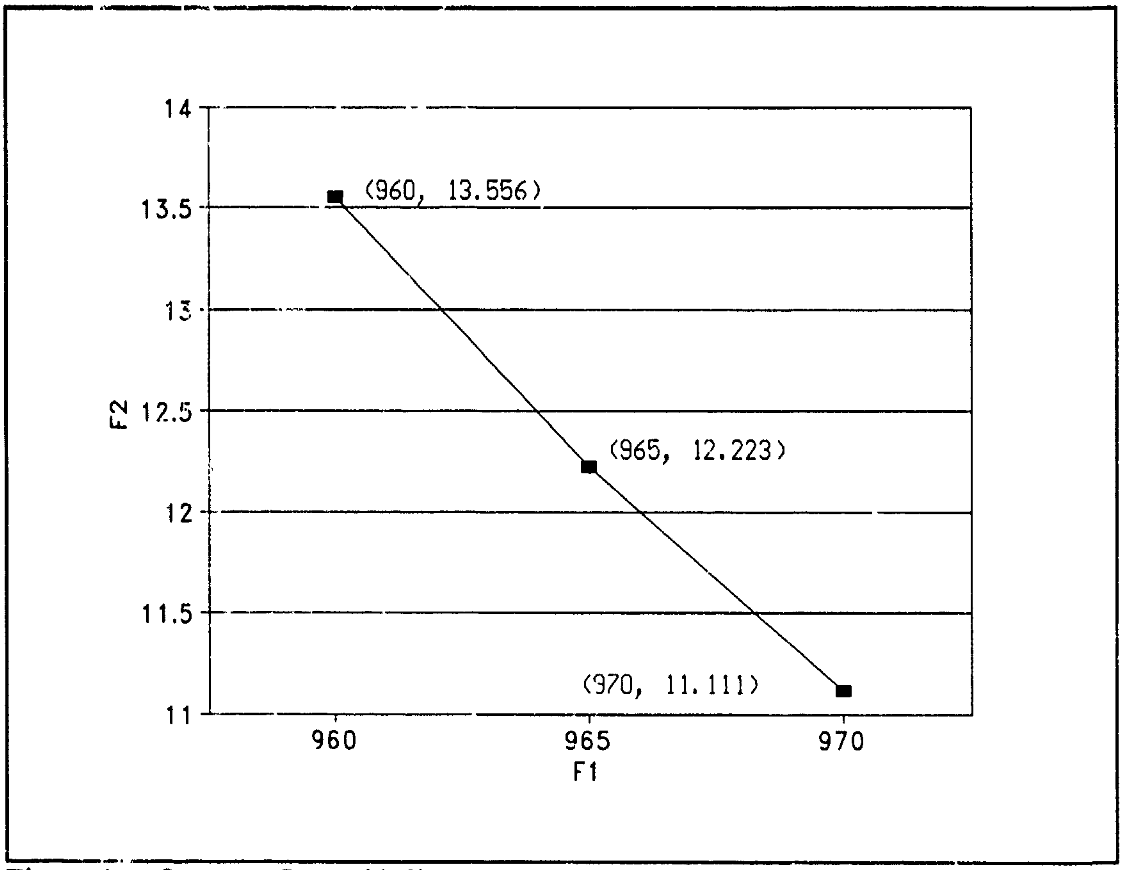


Figure 13 - Outcome Space (d=2)

Step 1. Generate all possible combinations. Set an upper limit of p=3 on the number of sites selected. (this restriction is based on the original question asked by NGBHQ which request 1-site, 2-site, and 3-site configurations).

Step 2. Determine the alternative space. For each possible alternative X^n, verify that:

$$\sum_{i}^{I} \sum_{j}^{J} X_i \, W_{ijk} \geq d_k \quad \forall J$$

where X_i is 0-1 integer

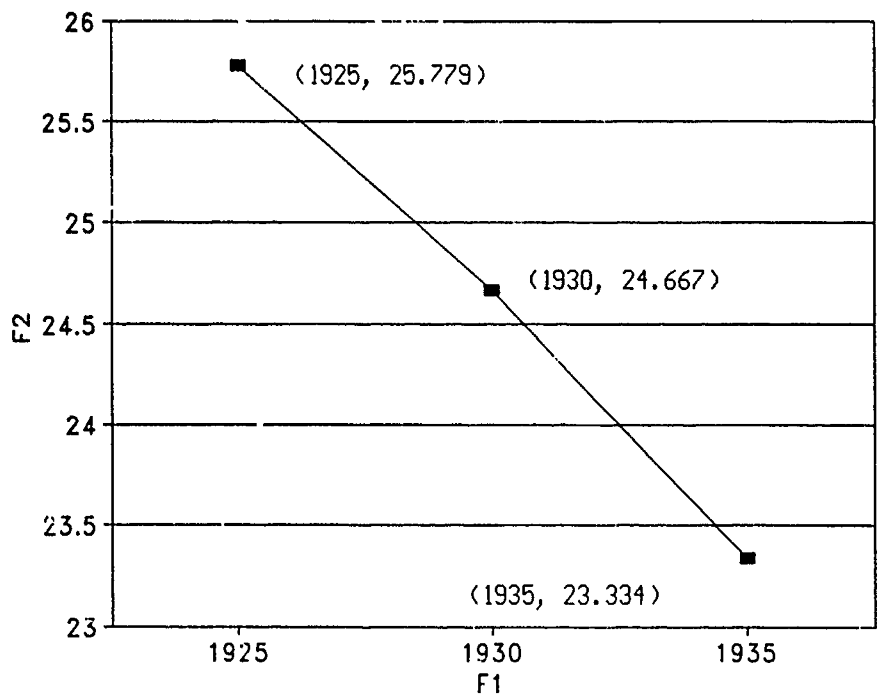


Figure 14 - Outcome Space (d=4)

Step 3. Map each point in the alternative space to the outcome space using the variance criterion function F2, and the following function for F1

Step 4. Determine the set of N-points by testing each point in the outcome space using the condition as described earlier in the maximal coverage exhaustive list algorithm.

$$F1 = \sum_{i}^{I} 1000X_i - (\sum_{i}^{I} \sum_{j}^{J} \sum_{k}^{K} X_i W_{ijk} - JKd_k)$$

SATISFICE LEVEL	F1	X^*
36.8900	1925	Sites 1,2
25.7790	1925	Sites 1.2
25.2900	1930	Sites 2,3
24.8010	1930	Sites 2,3
24.3120	1935	Sites 1,3
23.8230	1935	Sites 1,3
23.3340	1935	Sites 1,3

Table 13 - Multicriteria Set Covering Solutions (d=4)

X^*	r(y;1)	r(y;2)	r(y;∞)
Site 2	2.445	2.445	2.445
Site 1	6.112	5.122	5
Site 3	10.000	10.000	10.000

Table 14 - Norms (d_k=2)

X^*	r(y;1)	r(y;2)	r(y;∞)
Sites 1,2	2.445	2.445	2.445
Sites 2,3	6.330	5.175	5
Sites 1,3	10.000	10.000	10.000

Table 15 - Norms (d_k=4)

2.7 Compromise Solutions. As with the maximal coverage problem, we assume the weights are equal and investigate the impact of varying the parameter p on the ranking of the alternatives. The results for the $d_k=2$ problem are shown in Table 14.

The results for the $d_k=2$ problem are identical to the p=1 problem of the maximal coverage problem. This is a coincidence, since the set of N-points in the $d_k=2$ problem is governed by the requirements, d_k, of the demand points. Also, as was the case with the maximal coverage problem, varying the parameter p does not result in varying the ranking of the alternatives for this set of data. Similar results were obtained for the $d_k=4$ problem and these are shown in Table 15.

3. *A FORTRAN-based GEODSS Optimal Location Solver*

The end-product of the foregoing discussions, both in this and in the preceding chapter, has been the development of a FORTRAN-based solution algorithm for the GEODSS location problem of this research. The source code for this FORTRAN solver is included in Appendix U.

This FORTRAN program is written in modular form, with a separate subroutine to perform each major segment of the solution algorithm. While this technique may not be the most efficient from a memory usage perspective, it does present two distinct advantages. First, the flow of the solution methodology is easily recognized by reading through the main program which primarily consists of a series of calls to subroutines. Secondly, the modular design allows for easier code verification and modification functions since each subroutine can be easily compiled and executed seperately.

The FORTRAN program solves the multicriteria maximal coverage p=1, p=2, and p=3 problems and produces an ordered list of alternatives as solution to each of these problems. These ordered lists provide a de facto solution to the set covering problem, has the solution to the set covering problem is simply the top ranked alternative which minimizes p.

Extensive internal documentation is provided throughout the code. Additional comments are provided here to further assist the reader.

3.1 Input Files. As described in Chapter III, the input to the program consists of the following input files (see Appendix V):

PROBA.DAT	- 12x12 matrix of probability of event A, at location I, in month J
PROBB.DAT	- 12x12 matrix of probability of event B given A, at location I, in month J
PROBC.DAT	- 12X12 matrix of probability of event C given AB, at location I, in month J
PROBD.DAT	- 12x6 matrix of probability of event D, at location I, for satellite K
PROBE.DAT	- six 12x12 matrices of probability of event E given ABCD, at location I, in month J, for each satellite
PROBF.DAT	- six 12x12 matrices of probability of event F given AD, at location I, in month J, for each satellite
OBSREQ.DAT	- 1x6 vector of monthly requirement of observations for each satellite

where

A ≡ sun < 6 degrees below horizon
B ≡ wind speed < 25 knots
C ≡ temperature > -50 C
D ≡ satellite > 15 degrees above horizon
E ≡ CFLOS for 5 minutes
F ≡ satellite illuminated by the sun

3.2 Output Files. The output consists of eight files as follows:

RESULTS.OUT	- three tables of top 12 alternatives ordered by deviation from the ideal, and a table of E_{ij} for each of the six satellites. (see Appendix W)
ALTLST.OUT	- table showing alternative numbering legend used by the software
ALTOBS.OUT	- six 298x12 tables showing number of observations collected by alternative X, in month J, on each satellite

OBJFCN.OUT	- table showing value of F_1 and F_2 for each alternative
UTILS.OUT	- table showing utility of F_1 and F_2 for each alternative
FEASIB.OUT	- table showing the feasibility status (true or false) of each alternative
EFFSET.OUT	- table indicating whether or not an alternative is part of the non-dominated set
DEVIAT.OUT	- table showing the Manhattan metric deviation from the ideal for each alternative

3.3 Algorithm. The main program consists of calls to ten different subroutines. The only other executable statements in the main program are the initial assignments statements to the BLOCK vector as shown in Appendix U. The subprograms are executed in the following order:

(1) Subroutine PROCAL reads the probability data files and computes PROB(I,J,K), the probability that a 5 minute block has met all the necessary conditions for observing

(2) Subroutine EXPVAL computes EXPECT(I,J,K), the expected value of the number of 5 minute blocks usable for observation

(3) Subroutine FEACHK reads the observation requirements file and determine which alternatives are feasible by verifying whether or not an alternative provides the minimum requirement in each month of the year

(4) Subroutine OBJCAL calculates the value of both criterion functions for all feasible alternatives. This provides the outcome space coordinates for all feasible alternatives.

(5) Subroutine EFFSET determine which alternatives are non-dominated by comparing each outcome against all other outcomes in the outcome space.

(6) Subroutine IDLCAL computes the outcome space coordinates of the ideal (as defined in Chapter III).

(7) Subroutine UTILS converts the criterion function values to utils, by assigning a utility of 0 to the lowest OBJ1 value and the highest OBJ2 value, and a utility of 1 to the highest OBJ1 and the lowest OBJ2 values. A linear utility function is assumed, and utilities between 0 and 1 are assigned to all other feasible alternative criterion function values.

(8) Subroutine DEVCAL computes the deviation from the ideal in utils.

(9) Subroutine PRIORI orders the alternatives for each subproblem (p=1, p=2, p=3)

(10) Subroutine PRTOUT prints a summary of the findings to the primary output file RESULTS.OUT.

3.4 Assumptions and Limitations. As already stated above, the algorithm assumes a linear utility function for each criterion function. Furthermore, it is assumed that the weights placed on the criterion function are equal. Interviews with the decision maker revealed that these assumptions are not completely groundless (5). However, it would not be very difficult to modify the program to produce alternative solutions based on differing criterion function weights. This can be accomplished in subroutine DEVCAL by multiplying the objective function values by their respective weights when calculating the deviation from the ideal (50:70).

A serious limitation of the program is the lack of sensitivity analysis. This leaves unanswered many questions regarding the impact of varying input parameters on the final selection. For example, how much of a change in the P(CFLOS) at a given location is required before it is discarded as the optimal alternative?

On the positive side, the program provides an efficient way of sifting through masses of data, and allows the decison maker to bring together meteorological, astrodynamic, and operational factors in an analytical manner to arrive at a solution which is defensible from a MCDM theoretical standpoint.

4. *Conclusions*

The research has revealed the deficiency of the weighted sum approach in generating the set of N-points for the integer problem. The constraint method's ability to generate the complete set of N-points is limited by the number of steps that are taken in incrementing the constraint level of the second objective function.

In the maximal coverage problem, with $i=3$ candidate locations, the number of alternatives is limited to 3 in both the $p=1$ and $p=2$ problems. In the larger problem, with $i=12$ locations, the alternative space includes 12 points in the $p=1$ problem, $12!/2!10!=66$ points in the $p=2$ problem, and $12!/3!9!=220$ points in the $p=3$ problem. These quantities are manageable and the alternative and outcome spaces can be readily generated using the exhaustive-list algorithm described in this chapter.

The variance criterion function as expressed in this chapter seeks to minimize the variance in the number of observation opportunities provided to individual satellites from location i in month j. For solutions where the number of facilities selected is greater than one ($p>1$), another way of expressing the variance criterion function is to minimize the variance in the number of observations provided by alternative x in month j. The difference between the two is that in the latter case, single site coverage is not penalized as long as another site in the alternative set provides an "average" number of observations. This represents two different views of the problem and the decision maker needs to determine which view should be adopted. In the FORTRAN software developed for this research, the "alternative-based" variance criterion function is used.

VI. CASE STUDY

1. The Set of Candidate Locations

The number of candidate locations for this example was limited to the 12 sites listed at Table 16, which are also shown at Figure 15:

Site No.	Location	Lat (N)	Long (W)
1	Sandspit	53 15	131 49
2	Churchill	58 45	94 05
3	Penticton	49 28	119 36
4	Chatham	47 01	65 27
5	Torbay	47 38	52 42
6	Alert	82 30	62 20
7	Frobisher	63 45	68 34
8	Inuvik	68 18	133 29
9	London	43 02	81 09
10	Moose Jaw	50 20	105 33
11	Whitehorse	60 42	135 07
12	Cold Lake	54 25	110 17

Table 16 - Candidate Locations

With 12 sites, the major Canadian climatological regions described in Chapter II are well represented. It is assumed that the meteorological parameters of interest (cloud cover, wind speed, and temperature) do not greatly vary within these regions. Furthermore, the candidate location representing a given climate region should be

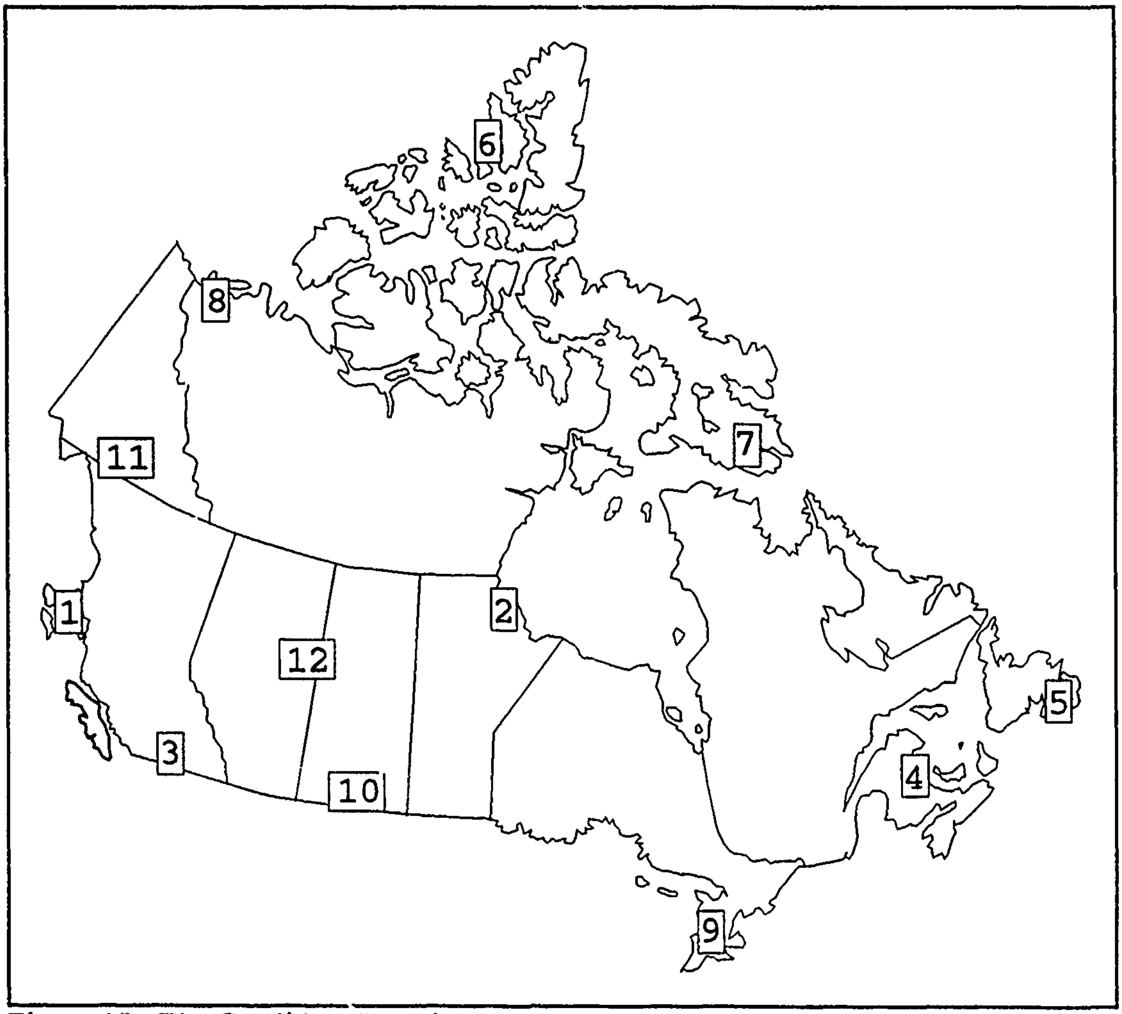


Figure 15 - The Candidate Locations

chosen so that the weather associated with this location is typical for the area in question. Therefore, the optimal alternative points to an area rather than a point on the map. A group of 12 sites also provides adequate geographic distribution to solve a 1-site, 2-site, and 3-site problem.

More sites would be needed if a study of Canadian climate revealed that the weather in the climate region represented by a candidate locations varied significantly. When solving the actual problem, it is therefore recommended that NDHQ meteorologists be consulted to determine whether or not more candidate sites are needed to provide a more complete climatic representation. However, the software provided with this research is designed for a set of 12 candidate sites. Increases in the number of sites will cause a factorial increase in required computer memory and run time and may require alterations to the software to improve its efficiency so that it can handle the larger problem. Taken to the extreme, a large enough increase in the number of candidate sites would require the development of an entirely new solution algorithm.

This example assumes that candidate locations have been pre-screened to ensure that they meet a number of essential criteria. First, historical weather data must be available for the chosen location. This usually means that there is an airfield located there, and therefore lists of Canadian airfields are of great help in selecting candidate sites. Second, a candidate location must meet a minimum standard of optical transmissivity (as discussed in Chapter II). For example, site surveys were performed to screen the current GEODSS locations by scientist from the Lincoln Laboratories of MIT (5). The technical details of the types of measurements performed in such surveys were not obtained, but these should include an evaluation of ambient lighting and particulate content of the atmosphere to ascertain whether or not the measured levels would interfere with GEODSS operations.

There may be other essential criteria that should be considered in the pre-screening of locations. However, as a result of interviews with the decision maker, it is felt that only the two criterion functions included in the location model fell into the "more is better" category (5).

2. *Choosing a Representative Satellite Population*

The satellite population of interest is defined as those objects "from 5000 kilometers altitude to geosynchronous altitudes in the western hemisphere", and of "sufficient brightness to be visible to the surveillance sensor" (18). For the purposes of this example, 6 fictitious satellites are selected:

Satellite 1 : a semi-synchronous satellite at an inclination of 65 degrees

Satellite 2 : a geostationary satellite at 50° West

Satellite 3 : " " at 70° West

Satellite 4 : " " at 100° West

Satellite 5 : " " at 120° West

Satellite 6 : " " at 140° West

For the actual problem, all synchronous satellites in inclined orbits should be included in the model to ensure that the chosen alternative provides coverage of these orbits. Since the orbital periods for these orbits is synchronized with the rotation of the earth, some locations will sometimes never "see" these satellites during an observation period. It is probably not necessary to include all the geostationary satellites of interest. This can be determined by looking at the limits of visibility for the candidate locations. However, the most easterly and westerly satellites that must be observed on the geostationary belt must be included.

The software provided with this research will support a large number of satellites. However, from a data gathering perspective, it is better to minimize the number of satellites in the solution procedure. To this end, it is asserted without rigorous astrodynamical proof that the alternative that provides optimal coverage of the synchronous population will also provide optimal coverage of the non-synchronous satellites. This assertion seems logical since these non-synchronous satellites will eventually pass over all points on earth at varying times of the day and should, therefore, be observable at some point in time. Therefore, it appears non-synchronous satellites can be safely excluded from the model.

One potentially serious limitation of the model is that elliptical orbits cannot be included at this point due to the fact that the ETAC CFLOS model is limited to determining the P(CFLOS) to satellites in circular orbits. Thus, elliptical synchronous orbits (such as the Molniya) are not included in this numerical example. To work around this problem, ETAC might be able to provide P(CFLOS) given a frequency distribution of elevation angles at a given location to a satellite in an elliptical orbit. Alternatively, the ETAC model might be upgraded to include elliptical orbits (48). However, before this additional developmental effort is undertaken, an impact analysis of leaving out the synchronous elliptical orbits on the selection of the optimal alternative should be performed.

3. *Feasibility Check of Geostationary Satellite Coverage*

By calculating the limits of visibility of the geostationary belt from each candidate location, it is possible to have a quick verification of whether or not a 1-site, 2-site, or 3-site solution exist. The plot in Figure 15 shows that the 1-site problem is infeasible just on the basis of covering the five geostationary satellites. If a 1-site solution is desired, than additional candidate locations should be selected near the midpoint (in longitude) of the required geostationary coverage.

4. *Data Collection and Input Files*

The input data for this numerical example is shown in Appendix V. The data is arranged in seven different input files. The PROB files represent the event probabilities as described earlier in Chapter V. The FORTRAN format of these files is as follows:

PROBA.DAT 12 single spaced rows of 12 columns each formatted as 12F4.2 (location by month)

PROBB.DAT 12 single spaced rows of 12 columns each formatted as 12F4.2 (location by month)

PROBC.DAT 12 single spaced rows of 12 columns each formatted as 12F4.2 (location by month)

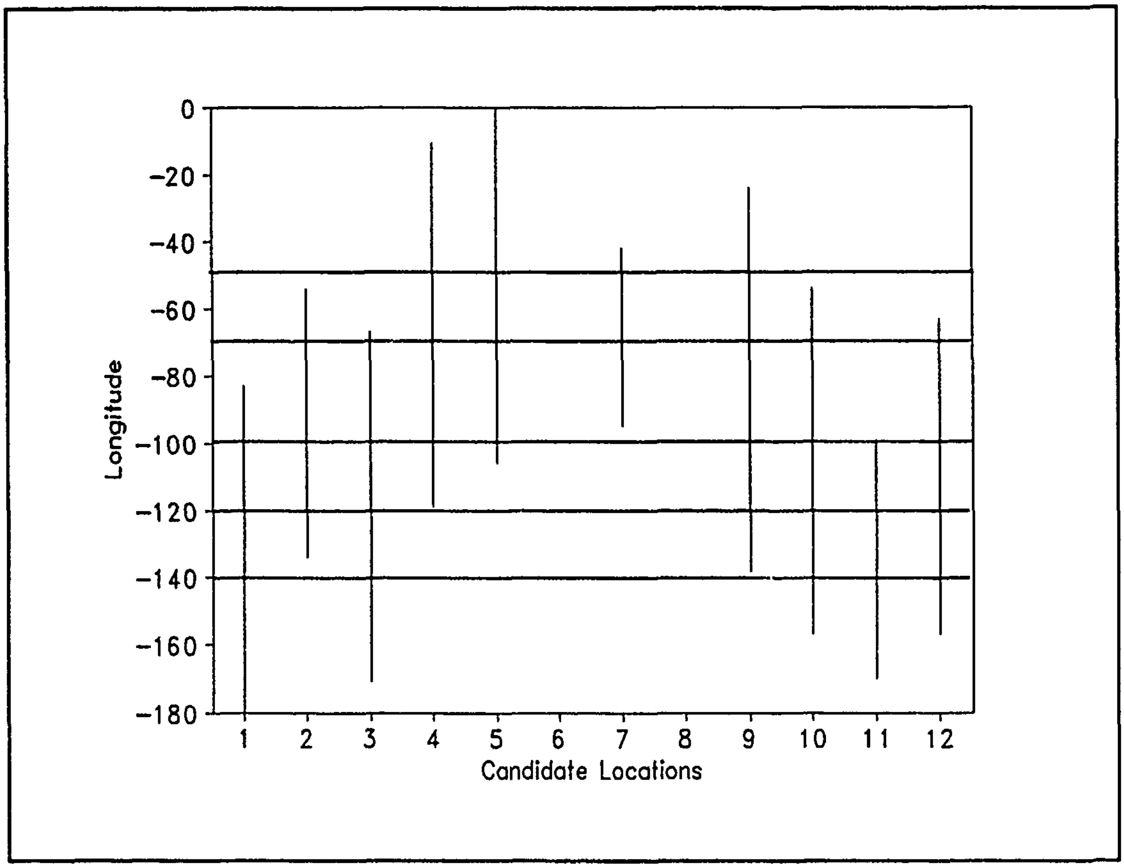


Figure 16 - Limits of Visibility

PROBD.DAT 12 single spaced rows of 6 columns each formatted as 6F4.2 (location by satellite)

PROBE.DAT 6 consecutive matrices of 12 single spaced rows with 12 columns formatted as 12F4.2 - A character string (e.g., Satellite 1) must be inserted before the first table and between all other tables

PROBF.DAT 6 consecutive matrices of 12 single spaced rows formatted exactly the same way as the PROBE.DAT file

As explained earlier in Chapter III, the PROBF.DAT data which represents P(F|AD) is obtained from P(FAD)/P(AD). This probability is calculated by dividing the number of minutes event FAD occurs by the number of minutes event AD occurs for each month of the year, for each location, and for each satellite. As can be seen from the probability files, the P(F|AD) is binary during non-geostationary eclipse months, and is .91 or .97 during four months of the year where the satellite is in fact situated above the 15 degree elevation threshold. The data for the semi-synchronous satellite (satellite 1) shows similar seasonal trends.

All PROBF.DAT data was obtained by running "pass scheduler" software developed at AFIT by Maj Kelso (22). All other probability data was graciously provided by ETAC (48).

The OBSREQ file is a row vector representing the monthly requirement of observations that must be collected on each satellite to maintain an accurate record of the orbital parameters. These requirements, which can be obtained from AFSPACECOM, are typically in the order or 30 observations/month (26). The row vector is formatted as 6F5.0.

5. *Analysis of Results*

The results are shown in Appendix W, which includes the RESULTS.OUT , ALTLST.OUT, FEASIB.OUT, AND EFFSET.OUT files. The other output files, namely ALTOBS.OUT, OBJFCN.OUT, DEVIAT.OUT, and UTILS.OUT, are not included with this report. These are quite large and do not necessarily need to be reviewed by the user to obtain an appreciation of the findings.

The ALTLST.OUT file provides a legend of the alternative numbers assigned to the various possible combinations of sites used throughout the program. The twelve p=1 alternatives use alternative numbers 1 to 12, the sixty-six p=2 alternatives use numbers 13 to 78, and the two hundred and twenty p=3 alternatives use numbers 79 to 298.

The RESULTS.OUT file shows that there are no feasible p=1 alternatives. As shown in Figure 15, this is due to the fact that none of the sites provides coverage of all five geostationary satellites. This is not to say that there does not exist a location in central Canada that would provide such coverage.

The output also shows that there are 12 feasible p=2 alternatives. Ten of these feasible alternatives are non-dominated (in N-set). The ordered list of feasible alternatives is repeated below at Table 17, along with the sites represented by the

ALT NO	SITE NO	Locations
15	1-4	Sandspit, Chatham
55	5-10	Torbay, Moose Jaw
35	3-5	Penticton, Torbay
57	5-12	Torbay, Cold Lake
39	3-9	Penticton, London
20	1-9	Sandspit, London
75	9-12	London, Cold Lake
48	4-10	Chatham, Moose Jaw
16	1-5	Sandspit, Torbay
73	9-10	London, Moose Jaw
50	4-12	Chatham, Cold Lake
34	3-4	Penticton, Chatham

Table 17 - Ordered Alternatives (p=2)

alternative numbers. It is immediately noted that none of the northerly locations are included in the list of feasible alternatives. This is probably due in part to the fact that the long periods of daylight in the summer and long winter nights introduce large variances during these seasons. Also, northerly locations afford very poor coverage of the geostationary belt and would rarely afford the necessary coverage in combination with only one southerly location.

We also note that none of the feasible pairs are composed of adjacent sites, and that generally each pair of sites includes a location in both eastern and western Canada. There are at least two reasons why this occurs. First, the fact that geostationary belt was populated with eastern and western satellites forces a solution that includes east and west sites. Secondly, it is known that the P(CFLOS) increases with increasing elevation angles (47). Therefore, the optimal location should also be the pair of sites that maximize the elevation angle to the satellite population.

Of final note, it is interesting to see that the primary alternative includes Chatham, New Brunswick. This happens to be the location of the Canadian Baker-Nunn satellite tracking station which is currently operational as an optical (film) satellite tracking system. However, while the Cold Lake site was at one time also active as a Baker-Nunn site, the Chatham-Cold Lake combination is ranked near the bottom at number 11.

The program found that 90 of the 220, p=3 alternatives were feasible. Of these, only 23 were non-dominated. This is an important fact, since if the assumption of equal weights for the criterion functions was put in doubt, and new weights were determined, the optimal 3-site alternative must by definition be one of the 23 non-dominated solutions. This means that only about 10% of the total number of alternatives would need to be scrutinized to determine which alternative is optimal.

As with the p=2 problem, the feasible alternatives were ordered by ascending order of deviation to the ideal. The ordered list is shown in Appendix W and repeated in Table 18 which also shows the site names. A marked difference between the p=2 and p=3 alternative set is that the 3-site alternatives favor the northerly locations while the 2-site solutions do not. All four northerly locations (Alert, Inuvik, Frobisher, and Whitehorse) appear at least once in the top 12 alternatives. In fact, a northerly location is included in all but three of the top 12 alternatives.

ALT NO	SITES	Locations
273	6-9-10	Alert, London, Moose Jaw
103	1-4-10	Sandspit, Chatham, Moose Jaw
283	7-9-10	Frobisher, London, Moose Jaw
185	3-4-11	Penticton, Chatham, Whitehorse
89	1-3-4	Sandspit, Penticton, Chatham
240	4-10-11	Chatham, Moose Jaw, Whitehorse
120	1-7-9	Sandspit, Frobisher, London
289	8-9-10	Inuvik, London, Moose Jaw
105	1-4-12	Sandspit, Chatham, Cold Lake
225	4-6-10	Chatham, Alert, Moose Jaw
275	6-9-12	Alert, London, Cold Lake
242	4-11-12	Chatham, Whitehorse, Cold Lake

Table 18 - Ordered Alternatives (p=3)

For example, the optimal alternative of the p=3 problem (Alt No. 273) includes Alert and London, which are the most northerly and southerly locations of the set of candidate set. The Alert-London pair appears again at alternative number 275. Further scrutiny of the ordered list shows that whenever London is chosen, the alternative also includes a northerly site. This occurs in 5 of 12 cases. The list also shows that an alternative is never composed of two northerly sites.

The popularity of northerly sites for the p=3 problem may be partly e-xplained by the fact that the 3-site solutions can take advantage of the long winter observing period provided by northerly locations as long as a pair of southerly sites are available in the summer months to provide the necessary observations. A minimum of two southerly sites are required to provide coverage of the geostationary belt.

An additional advantage in locating north can be seen by inspection of the RESULTS.OUT table entitled "Expected Number of Observations for Sat 1". This table shows that, in winter months, the northerly locations provide better coverage of the synchronous orbits. This is due to the obvious fact that the chances of seeing a near-polar orbiting satellite increase as you increase your latitude (north or south).

The optimal solution to the GEODSS set covering problem is found simply by selecting the top ranked alternative from the list with the smallest "p" value. For this example, the optimal set covering alternative is Alternative 15.

VII. CONCLUSION

1. Summary

This research has presented a study of the maximal coverage p-median and of the set covering facility location problems as applied to the GEODSS facility location problem. The classical single-objective mathematical formulations of both of these problems were converted into network-flow formulations and various solution methodologies were developed using a scaled-down version of the GEODSS problem.

The next step of the research was the introduction of a second criterion function into the problem. This second function consisted of minimizing the sum of the variance in coverage at the selected locations. The research revealed the deficiencies of MOLP (multiobjective linear programming) techniques in generating the efficient frontier of an integer problem. A "brute-force" solution algorithm was developed and coded in FORTRAN 77 that generates all feasible alternatives, determines which of these are non-dominated, and then provides an ordered list using paired comparisons with the ideal.

A numerical example was presented which showed the difficulty in finding a feasible one-site solution given the need to observe a wide segment of the geostationary belt. The example also showed that, for a given satellite population, the optimal alternative must for similar reasons include two southerly locations. The example revealed that while two-site solutions therefore exclude northerly locations, three-site solutions will usually include a northerly location.

2. *Conclusions*

While the numerical example presented in Chapter 6 included only a limited set of satellites, and excluded the synchronous elliptical Molniya orbit, two tentative conclusions can be drawn pending analysis of new data which would hopefully include the Molniya and additional sites.

First, coverage of the geostationary belt is a dominating force in the selection of the optimal alternative. This coverage probably needs to be accomplished by a minimum of two southerly locations.

Second, if a third site is added to this pair, then it seems reasonable to want to locate north, especially if highly inclined synchronous orbits are a part of the satellite population.

3. *Recommendations for Further Study*

The GEODSS location problem is solved by making use of the concepts and theories of three distinct disciplines. These are the fields of operations research, astrodynamics, and meteorology. The solution methodology proposed in this research could be improved by further work in any or all three of these areas.

A major limitation of the solution methodology is the lack of sensitivity analysis. A procedure needs to be devised to determine the sensitivity of the solution to the input parameters.

If the model needs to be applied to problems involving a larger number of locations, then a multicriteria integer solver algorithm, perhaps based on the Branch and Bound algorithm, could be developed.

The effects of the moon on the GEODSS are not modelled in this research. This large source of light in the night sky negates GEODSS tracking of satellites within 15 degrees of the full moon (14). It is debatable whether or not lunar effects need to be included in the model. Since operations can continue in other parts of the sky, the problem with the moon might be dismissed by scheduling passes at other times. However, the model presented here can easily accept an additional term in the probability equation to take into account the effect of the moon.

The amount of sunlight reflected by a satellite is a function of the solar phase angle. Essentially, this angle provides an indication of what portion of the satellite is illuminated (17:23). The effect is similar to the various phases of the moon. It must be determined whether or not, for a given orbit, the distribution of reflected solar flux from a satellite varies from location to location.

The model does not reward redundant coverage (simultaneous coverage of a satellite by more than one site). Redundancy of coverage is an important operational consideration that is invoked to ensure continued observation flow in spite of isolated site failures. The decision to provide redundancy involves a tradeoff between cost and operational feasibility. Consideration might be given to providing for redundancy of coverage by obtaining a mobile or relocatable system. For example, in the 3-site solutions which involve northern sites, the northerly locations could be equipped with relocatable systems.

Meteorological considerations are a key part of this problem. More in-depth study of the Canadian climate is required to obtain a complete set of regions of homogeneous climatology. Within each of these regions, a representative location with historical weather data is chosen, and probability tables generated. These could then be input in the FORTRAN program provided herein assuming the final set is not too large.

The ETAC CFLOS model might be upgraded to include an algorithm to generate positions of synchronous, and elliptical orbits. Alternatively, a way of obtaining estimates of the P(CFLOS) from a given location based on the frequency distribution of the elevation angle to the satellite might be developed.

```
..TITLE
     MAXIMAL COVERAGE MIP83 FORMULATION (p=1)
..OBJECTIVE MAXIMIZE
*source connectors
 0 [[XS1]] + 0 [[XS2]] + 0 [[XS3]]
*state 1
+ 8 [[X110]] + 5 [[X111]] + 4 [[X112]]
+ 3 [[X210]] + 6 [[X211]] + 9 [[X212]]
+ 2 [[X310]] + 7 [[X311]] + 5 [[X312]]
*state 2
+ 6 [[X413]] + 2 [[X414]] + 7 [[X415]]
+ 9 [[X513]] + 3 [[X514]] + 6 [[X515]]
+ 4 [[X613]] + 6 [[X614]] + 5 [[X615]]
*state 3
+ 4 [[X716]] + 8 [[X717]] + 9 [[X718]]
+ 6 [[X816]] + 7 [[X817]] + 9 [[X818]]
+ 3 [[X916]] + 9 [[X917]] + 7 [[X918]]
*interstate links
+ 0 [[X14]] + 0 [[X25]] + 0 [[X36]]
+ 0 [[X47]] + 0 [[X58]] + 0 [[X69]]
*sink connectors
+ 0 [X10T] + 0 [X11T] + 0 [X12T]
+ 0 [X13T] + 0 [X14T] + 0 [X15T]
+ 0 [X16T] + 0 [X17T] + 0 [X18T]
*select 'p' facilities
```

+ 0 p

..CONSTRAINTS

```
NODE  S: XS1 + XS2 + XS3 - p = 0
NODE  1: X110 + X111 + X112 + X14 - 4 XS1 = 0
NODE  2: X210 + X211 + X212 + X25 - 4 XS2 = 0
NODE  3: X310 + X311 + X312 + X36 - 4 XS3 = 0
NODE  4: X413 + X414 + X415 + X47 - 4 X14 = 0
NODE  5: X513 + X514 + X515 + X58 - 4 X25 = 0
NODE  6: X613 + X614 + X615 + X69 - 4 X36 = 0
NODE  7: X716 + X717 + X718 - 3 X47 = 0
NODE  8: X816 + X817 + X818 - 3 X58 = 0
NODE  9: X916 + X917 + X918 - 3 X69 = 0
NODE 10: X10T - X110 - X210 - X310 = 0
NODE 11: X11T - X111 - X211 - X311 = 0
NODE 12: X12T - X112 - X212 - X312 = 0
NODE 13: X13T - X413 - X513 - X613 = 0
NODE 14: X14T - X414 - X514 - X614 = 0
NODE 15: X15T - X415 - X515 - X615 = 0
NODE 16: X16T - X716 - X816 - X916 = 0
NODE 17: X17T - X717 - X817 - X917 = 0
NODE 18: X18T - X718 - X818 - X918 = 0
*-----------------------------------------------------*
* for the next constraint, coefficient of 'p' is (j*k)
* where
*       j = number of states (months)
*       k = number of demand locations (missions)
*-----------------------------------------------------*
NODE  T: - X10T - X11T - X12T
```

- X13T - X14T - X15T
- X16T - X17T - X18T + 9 p = 0

* select 'p' facilities
p = 1

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MAXIMIZED.
MIP Strategy: 1
Variables: 46
Integer: 45
Constraints: 21
0 LE, 21 EQ, 0 GE.
Non-zero LP elements: 93
Disk Space: 0K bytes.
Page Space: 8K bytes.
Capacity: 7.7% used.
Estimated Time: 00:00:11

Iter 44
Solution Time: 00:00:01
May have A L T E R N A T E S O L U T I O N

Optimal Solution: 74.7778 Max Node Depth: 590 Limit: NONE

Solution: 58.0000 Iter: 51 Nodes: 7 Iteration Time: 00:00:08
INTEGER SOLUTION

File: Form 8/12/90 10:29:42 Page 1-1
SOLUTION (Maximized): 58.0000 OPER767 PROJECT 1

Variable	Activity	Cost	Variable	Activity	Cost
XS1	0.0000	0.0000	XS2	1.0000	0.0000
XS3	0.0000	0.0000	X110	0.0000	8.0000
X111	0.0000	5.0000	X112	0.0000	4.0000
X210	1.0000	3.0000	X211	1.0000	6.0000
X212	1.0000	9.0000	X310	0.0000	2.0000
X311	0.0000	7.0000	X312	0.0000	5.0000
X413	0.0000	6.0000	X414	0.0000	2.0000
X415	0.0000	7.0000	X513	1.0000	9.0000
X514	1.0000	3.0000	X515	1.0000	6.0000
X613	0.0000	4.0000	X614	0.0000	6.0000

File: Form 8/12/90 10:29:42 Page 1-2
SOLUTION (Maximized): 58.0000 OPER767 PROJECT 1

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.0000	5.0000	X716	0.0000	4.0000
X717	0.0000	8.0000	X718	0.0000	9.0000
X816	1.0000	6.0000	X817	1.0000	7.0000
X818	1.0000	9.0000	X916	0.0000	3.0000
X917	0.0000	9.0000	X918	0.0000	7.0000
X14	0.0000	0.0000	X25	1.0000	0.0000
X36	0.0000	0.0000	X47	0.0000	0.0000
X58	1.0000	0.0000	X69	0.0000	0.0000
X10T	1.0000	0.0000	X11T	1.0000	0.0000
X12T	1.0000	0.0000	X13T	1.0000	0.0000

File: Form 8/12/90 10:29:42 Page 1-3
SOLUTION (Maximized): 58.0000 OPER767 PROJECT 1

Variable	Activity	Cost	Variable	Activity	Cost
X14T	1.0000	0.0000	X15T	1.0000	0.0000
X16T	1.0000	0.0000	X17T	1.0000	0.0000
X18T	1.0000	0.0000	p	1.0000	0.0000

File: Form 8/12/90 10:29:42 Page 1-4
CONSTRAINTS: OPER767 PROJECT 1

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	0.0000 =	0.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000

NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	0.0000 =	0.0000

File: Form 8/12/90 10:29:42 Page 1-5
CONSTRAINTS: OPER767 PROJECT 1

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 21	1.0000 =	1.0000			

Total Error: 0.000000

..TITLE

MAXIMAL COVERAGE MIP83 FORMULATION (p=2)

..OBJECTIVE MAXIMIZE

*source connectors

0 [[XS1]] + 0 [[XS2]] + 0 [[XS3]]

*state 1

+ 8 [[X110]] + 5 [[X111]] + 4 [[X112]]

+ 3 [[X210]] + 6 [[X211]] + 9 [[X212]]

+ 2 [[X310]] + 7 [[X311]] + 5 [[X312]]

*state 2

+ 6 [[X413]] + 2 [[X414]] + 7 [[X415]]

+ 9 [[X513]] + 3 [[X514]] + 6 [[X515]]

+ 4 [[X613]] + 6 [[X614]] + 5 [[X615]]

*state 3

+ 4 [[X716]] + 8 [[X717]] + 9 [[X718]]

+ 6 [[X816]] + 7 [[X817]] + 9 [[X818]]

+ 3 [[X916]] + 9 [[X917]] + 7 [[X918]]

*interstate links

+ 0 [[X14]] + 0 [[X25]] + 0 [[X36]]

+ 0 [[X47]] + 0 [[X58]] + 0 [[X69]]

*sink connectors

+ 0 [X10T] + 0 [X11T] + 0 [X12T]

+ 0 [X13T] + 0 [X14T] + 0 [X15T]

+ 0 [X16T] + 0 [X17T] + 0 [X18T]

*select 'p' facilities

+ 0 p

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 - p = 0

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T - X110 - X210 - X310 = 0

NODE 11: X11T - X111 - X211 - X311 = 0

NODE 12: X12T - X112 - X212 - X312 = 0

NODE 13: X13T - X413 - X513 - X613 = 0

NODE 14: X14T - X414 - X514 - X614 = 0

NODE 15: X15T - X415 - X515 - X615 = 0

NODE 16: X16T - X716 - X816 - X916 = 0

NODE 17: X17T - X717 - X817 - X917 = 0

NODE 18: X18T - X718 - X818 - X918 = 0

```
*-------------------------------------------------------*
* for the next constraint, coefficient of 'p' is (j*k)
* where
*       j = number of states (months)
*       k = number of demand locations (missions)
*-------------------------------------------------------*
```

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T + 9 p = 0

* select 'p' facilities
p = 2

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MAXIMIZED.
MIP Strategy: 1
Variables: 46
Integer: 45
Constraints: 21
0 LE, 21 EQ, 0 GE.
Non-zero LP elements: 93
Disk Space: 0K bytes.
Page Space: 8K bytes.
Capacity: 7.7% used.
Estimated Time: 00:00:11

Iter 53
Solution Time: 00:00:02
U N I Q U E S O L U T I O N

Optimal Solution: 128.6364 Max Node Depth: 590 Limit: NONE

Solution: 111.0000 Iter: 60 Nodes: 7 Iteration Time: 00:00:10
INTEGER SOLUTION

File: Form 8/12/90 10:27:32 Page 1-1
SOLUTION (Maximized): 111.0000 OPER767 PROJECT 1

Variable	Activity	Cost	Variable	Activity	Cost
XS1	1.0000	0.0000	XS2	1.0000	0.0000
XS3	0.0000	0.0000	X110	1.0000	8.0000
X111	1.0000	5.0000	X112	1.0000	4.0000
X210	1.0000	3.0000	X211	1.0000	6.0000
X212	1.0000	9.0000	X310	0.0000	2.0000
X311	0.0000	7.0000	X312	0.0000	5.0000
X413	1.0000	6.0000	X414	1.0000	2.0000

X415	1.0000	7.0000	X513	1.0000	9.0000
X514	1.0000	3.0000	X515	1.0000	6.0000
X613	0.0000	4.0000	X614	0.0000	6.0000

File: Form 8/12/90 10:27:32 Page 1-2
SOLUTION (Maximized): 111.0000 OPER767 PROJECT 1

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.0000	5.0000	X716	1.0000	4.0000
X717	1.0000	8.0000	X718	1.0000	9.0000
X816	1.0000	6.0000	X817	1.0000	7.0000
X818	1.0000	9.0000	X916	0.0000	3.0000
X917	0.0000	9.0000	X918	0.0000	7.0000
X14	1.0000	0.0000	X25	1.0000	0.0000
X36	0.0000	0.0000	X47	1.0000	0.0000
X58	1.0000	0.0000	X69	0.0000	0.0000
X10T	2.0000	0.0000	X11T	2.0000	0.0000
X12T	2.0000	0.0000	X13T	2.0000	0.0000

File: Form 8/12/90 10:27:32 Page 1-3
SOLUTION (Maximized): 111.0000 OPER767 PROJECT 1

Variable	Activity	Cost	Variable	Activity	Cost
X14T	2.0000	0.0000	X15T	2.0000	0.0000
X16T	2.0000	0.0000	X17T	2.0000	0.0000
X18T	2.0000	0.0000	p	2.0000	0.0000

File: Form 8/12/90 10:27:32 Page 1-4
CONSTRAINTS: OPER767 PROJECT 1

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	0.0000 =	0.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000

NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 –	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	0.0000 =	0.0000

File: Form **8/12/90 10:27:32 Page 1-5**

CONSTRAINTS: OPER767 PROJECT 1

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 21	2.0000 =	2.0000			

Total Error: 0.000000

```
..TITLE
MAXIMAL COVERAGE LP83 FORMULATION (p=1)
..OBJECTIVE MAXIMIZE
 *source connectors
  0 XS1  + 0 XS2  + 0 XS3
 *state 1
 + 8 X110 + 5 X111 + 4 X112
 + 3 X210 + 6 X211 + 9 X212
 + 2 X310 + 7 X311 + 5 X312
 *state 2
 + 6 X413 + 2 X414 + 7 X415
 + 9 X513 + 3 X514 + 6 X515
 + 4 X613 + 6 X614 + 5 X615
 *state 3
 + 4 X716 + 8 X717 + 9 X718
 + 6 X816 + 7 X817 + 9 X818
 + 3 X916 + 9 X917 + 7 X918
 *interstate links
 + 0 X14 + 0 X25 + 0 X36
 + 0 X47 + 0 X58 + 0 X69
 *sink connectors
 + 0  X10T  + 0  X11T  + 0  X12T
 + 0  X13T  + 0  X14T  + 0  X15T
 + 0  X16T  + 0  X17T  + 0  X18T
 *select 'p' facilities
```

+ 0 p

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 - p = 0

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T - X110 - X210 - X310 = 0

NODE 11: X11T - X111 - X211 - X311 = 0

NODE 12: X12T - X112 - X212 - X312 = 0

NODE 13: X13T - X413 - X513 - X613 = 0

NODE 14: X14T - X414 - X514 - X614 = 0

NODE 15: X15T - X415 - X515 - X615 = 0

NODE 16: X16T - X716 - X816 - X916 = 0

NODE 17: X17T - X717 - X817 - X917 = 0

NODE 18: X18T - X718 - X818 - X918 = 0

```
*-------------------------------------------------------*
* for the next constraint, coefficient of 'p' is (j*k)
* where
*       j = number of states (months)
*       k = number of demand locations (missions)
*-------------------------------------------------------*
```

```
NODE  T: - X10T - X11T - X12T
         - X13T - X14T - X15T
         - X16T - X17T - X18T + 9 p = 0

* select 'p' facilities
p = 1

* source connector flows
XS1 <= 1
XS2 <= 1
XS3 <= 1

* sink-connector flows
X10T - p = 0
X11T - p = 0
X12T - p = 0
X13T - p = 0
X14T - p = 0
X15T - p = 0
X16T - p = 0
X17T - p = 0
X18T - p = 0

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1  = 0
X110 - X14  = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2  = 0
X210 - X25  = 0


X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3  = 0
X310 - X36  = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47  = 0
X14  - X47  = 0

X513 - X514 = 0
X513 - X515 = 0
```

X514 - X515 = 0
X513 - X58 = 0
X25 - X58 = 0

X613 - X614 = 0
X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
LP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MAXIMIZED.
Variables: 46
Constraints: 75
3 LE, 72 EQ, 0 GE.
Non-zero LP elements: 198
Disk Space: 0K bytes.
Page Space: 28K bytes.
Capacity: 15.7% used.
Estimated Time: 00:00:39

Iter 44
Solution Time: 00:00:03
May have A L T E R N A T E S O L U T I O N

File: Form1 9/03/90 23:53:31 Page 1-1
SOLUTION (Maximized): 58.0000 OPER767 TERM PROJECT -

Variable	Activity	Cost	Variable	Activity	Cost
XS1	0.0000	0.0000	XS2	1.0000	0.0000
XS3	0.0000	0.0000	X110	0.0000	8.0000
X111	0.0000	5.0000	X112	0.0000	4.0000
X210	1.0000	3.0000	X211	1.0000	6.0000

X212	1.0000	9.0000	X310	0.0000	2.0000
X311	0.0000	7.0000	X312	0.0000	5.0000
X413	0.0000	6.0000	X414	0.0000	2.0000
X415	0.0000	7.0000	X513	1.0000	9.0000
X514	1.0000	3.0000	X515	1.0000	6.0000
X613	0.0000	4.0000	X614	0.0000	6.0000

File: Form1 9/03/90 23:53:31 Page 1-2
SOLUTION (Maximized): 58.0000 OPER767 TERM PROJECT -

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.0000	5.0000	X716	0.0000	4.0000
X717	0.0000	8.0000	X718	0.0000	9.0000
X816	1.0000	6.0000	X817	1.0000	7.0000
X818	1.0000	9.0000	X916	0.0000	3.0000
X917	0.0000	9.0000	X918	0.0000	7.0000
X14	0.0000	0.0000	X25	1.0000	0.0000
X36	0.0000	0.0000	X47	0.0000	0.0000
X58	1.0000	0.0000	X69	0.0000	0.0000
X10T	1.0000	0.0000	X11T	1.0000	0.0000
X12T	1.0000	0.0000	X13T	1.0000	0.0000

File: Form1 9/03/90 23:53:31 Page 1-3
SOLUTION (Maximized): 58.0000 OPER767 TERM PROJECT -

Variable	Activity	Cost	Variable	Activity	Cost
X14T	1.0000	0.0000	X15T	1.0000	0.0000
X16T	1.0000	0.0000	X17T	1.0000	0.0000
X18T	1.0000	0.0000	p	1.0000	0.0000

File: Form1 9/03/90 23:53:31 Page 1-4
CONSTRAINTS: OPER767 TERM PROJECT -

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	0.0000 =	0.0000	NODE 1	0.0000 =	0.0000

NODE 2	0.0000	=	0.0000	NODE 3	0.0000	=	0.0000
NODE 4	0.0000	=	0.0000	NODE 5	0.0000	=	0.0000
NODE 6	0.0000	=	0.0000	NODE 7	0.0000	=	0.0000
NODE 8	0.0000	=	0.0000	NODE 9	0.0000	=	0.0000
NODE 10	0.0000	=	0.0000	NODE 11	0.0000	=	0.0000
NODE 12	0.0000	=	0.0000	NODE 13	0.0000	=	0.0000
NODE 14	0.0000	=	0.0000	NODE 15	0.0000	=	0.0000
NODE 16	0.0000	=	0.0000	NODE 17	0.0000	=	0.0000
NODE 18	0.0000	=	0.0000	NODE T	0.0000	=	0.0000

File: Form1 9/03/90 23:53:31 Page 1-5

CONSTRAINTS: OPER767 TERM PROJECT -

Constraint	Activity		RHS	Constraint	Activity		RHS
Row 21	1.0000	=	1.0000	Row 22	0.0000	<	1.0000
Row 23	1.0000	<	1.0000	Row 24	0.0000	<	1.0000
Row 25	0.0000	=	0.0000	Row 26	0.0000	=	0.0000
Row 27	0.0000	=	0.0000	Row 28	0.0000	=	0.0000
Row 29	0.0000	=	0.0000	Row 30	0.0000	=	0.0000
Row 31	0.0000	=	0.0000	Row 32	0.0000	=	0.0000
Row 33	0.0000	=	0.0000	Row 34	0.0000	=	0.0000
Row 35	0.0000	=	0.0000	Row 36	0.0000	=	0.0000
Row 37	0.0000	=	0.0000	Row 38	0.0000	=	0.0000
Row 39	0.0000	=	0.0000	Row 40	0.0000	=	0.0000

File: Form1 9/03/90 23:53:31 Page 1-6

CONSTRAINTS: OPER767 TERM PROJECT

Constraint	Activity		RHS	Constraint	Activity		RHS
Row 41	0.0000	=	0.0000	Row 42	0.0000	=	0.0000
Row 43	0.0000	=	0.0000	Row 44	0.0000	=	0.0000
Row 45	0.0000	=	0.0000	Row 46	0.0000	=	0.0000
Row 47	0.0000	=	0.0000	Row 48	0.0000	=	0.0000
Row 49	0.0000	=	0.0000	Row 50	0.0000	=	0.0000

Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000
Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000
Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

File: Form1 **9/03/90 23:53:31 Page 1-7**

CONSTRAINTS: OPER767 TERM PROJECT -

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000
Row 67	0.0000 =	0.0000	Row 68	0.0000 =	0.0000
Row 69	0.0000 =	0.0000	Row 70	0.0000 =	0.0000
Row 71	0.0000 =	0.0000	Row 72	0.0000 =	0.0000
Row 73	0.0000 =	0.0000	Row 74	0.0000 =	0.0000
Row 75	0.0000 =	0.0000			

Total Error: 0.000000

```
..TITLE
    MAXIMAL COVERAGE LP83 FORMULATION (p=2)
..OBJECTIVE MAXIMIZE
 *source connectors
  0 XS1  + 0 XS2  + 0 XS3
 *state 1
 + 8 X110 + 5 X111 + 4 X112
 + 3 X210 + 6 X211 + 9 X212
 + 2 X310 + 7 X311 + 5 X312
 *state 2
 + 6 X413 + 2 X414 + 7 X415
 + 9 X513 + 3 X514 + 6 X515
 + 4 X613 + 6 X614 + 5 X615
 *state 3
 + 4 X716 + 8 X717 + 9 X718
 + 6 X816 + 7 X817 + 9 X818
 + 3 X916 + 9 X917 + 7 X918
 *interstate links
 + 0 X14 + 0 X25 + 0 X36
 + 0 X47 + 0 X58 + 0 X69
 *sink connectors
 + 0  X10T  + 0  X11T  + 0  X12T
 + 0  X13T  + 0  X14T  + 0  X15T
 + 0  X16T  + 0  X17T  + 0  X18T
 *select 'p' facilities
```

```
+ 0 p

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 - p = 0

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T - X110 - X210 - X310 = 0

NODE 11: X11T - X111 - X211 - X311 = 0

NODE 12: X12T - X112 - X212 - X312 = 0

NODE 13: X13T - X413 - X513 - X613 = 0

NODE 14: X14T - X414 - X514 - X614 = 0

NODE 15: X15T - X415 - X515 - X615 = 0

NODE 16: X16T - X716 - X816 - X916 = 0

NODE 17: X17T - X717 - X817 - X917 = 0

NODE 18: X18T - X718 - X818 - X918 = 0
*-----------------------------------------------------*
* for the next constraint, coefficient of 'p' is (j*k)
* where
*       j = number of states (months)
*       k = number of demand locations (missions)
*-----------------------------------------------------*

NODE T: - X10T - X11T - X12T
        - X13T - X14T - X15T
        - X16T - X17T - X18T + 9 p = 0
```

```
* select 'p' facilities
p = 2

* source connector flows
XS1 <= 1
XS2 <= 1
XS3 <= 1

* sink-connector flows
X10T - p = 0
X11T - p = 0
X12T - p = 0
X13T - p = 0
X14T - p = 0
X15T - p = 0
X16T - p = 0
X17T - p = 0
X18T - p = 0

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1  = 0
X110 - X14  = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2  = 0
X210 - X25  = 0


X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3  = 0
X310 - X36  = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47  = 0
X14  - X47  = 0

X513 - X514 = 0
X513 - X515 = 0
X514 - X515 = 0
X513 - X58  = 0
X25  - X58  = 0

X613 - X614 = 0
```

X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
LP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MAXIMIZED.
Variables: 46
Constraints: 75
3 LE, 72 EQ, 0 GE.
Non-zero LP elements: 198
Disk Space: 0K bytes.
Page Space: 28K bytes.
Capacity: 15.7% used.
Estimated Time: 00:00:39

Iter 47
Solution Time: 00:00:03
May have A L T E R N A T E S O L U T I O N

File: Form2 9/03/90 23:54:44 Page 1-1
SOLUTION (Maximized): 111.0000 OPER767 TERM PROJECT -

Variable	Activity	Cost	Variable	Activity	Cost
XS1	1.0000	0.0000	XS2	1.0000	0.0000
XS3	0.0000	0.0000	X110	1.0000	8.0000
X111	1.0000	5.0000	X112	1.0000	4.0000
X210	1.0000	3.0000	X211	1.0000	6.0000
X212	1.0000	9.0000	X310	0.0000	2.0000
X311	0.0000	7.0000	X312	0.0000	5.0000
X413	1.0000	6.0000	X414	1.0000	2.0000

X415	1.0000	7.0000	X513	1.0000	9.0000
X514	1.0000	3.0000	X515	1.0000	6.0000
X613	0.0000	4.0000	X614	0.0000	6.0000

File: Form2 9/03/90 23:54:44 Page 1-2
SOLUTION (Maximized): 111.0000 OPER767 TERM PROJECT -

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.0000	5.0000	X716	1.0000	4.0000
X717	1.0000	8.0000	X718	1.0000	9.0000
X816	1.0000	6.0000	X817	1.0000	7.0000
X818	1.0000	9.0000	X916	0.0000	3.0000
X917	0.0000	9.0000	X918	0.0000	7.0000
X14	1.0000	0.0000	X25	1.0000	0.0000
X36	0.0000	0.0000	X47	1.0000	0.0000
X58	1.0000	0.0000	X69	0.0000	0.0000
X10T	2.0000	0.0000	X11T	2.0000	0.0000
X12T	2.0000	0.0000	X13T	2.0000	0.0000

File: Form2 9/03/90 23:54:44 Page 1-3
SOLUTION (Maximized): 111.0000 OPER767 TERM PROJECT -

Variable	Activity	Cost	Variable	Activity	Cost
X14T	2.0000	0.0000	X15T	2.0000	0.0000
X16T	2.0000	0.0000	X17T	2.0000	0.0000
X18T	2.0000	0.0000	p	2.0000	0.0000

File: Form2 9/03/90 23:54:44 Page 1-4
CONSTRAINTS: OPER767 TERM PROJECT -

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	0.0000 =	0.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000

NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	0.0000 =	0.0000

File: Form2 9/03/90 23:54:44 Page 1-5
CONSTRAINTS: OPER767 TERM PROJECT -

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 21	2.0000 =	2.0000	Row 22	1.0000 <	1.0000
Row 23	1.0000 <	1.0000	Row 24	0.0000 <	1.0000
Row 25	0.0000 =	0.0000	Row 26	0.0000 =	0.0000
Row 27	0.0000 =	0.0000	Row 28	0.0000 =	0.0000
Row 29	0.0000 =	0.0000	Row 30	0.0000 =	0.0000
Row 31	0.0000 =	0.0000	Row 32	0.0000 =	0.0000
Row 33	0.0000 =	0.0000	Row 34	0.0000 =	0.0000
Row 35	0.0000 =	0.0000	Row 36	0.0000 =	0.0000
Row 37	0.0000 =	0.0000	Row 38	0.0000 =	0.0000
Row 39	0.0000 =	0.0000	Row 40	0.0000 =	0.0000

File: Form2 9/03/90 23:54:44 Page 1-6
CONSTRAINTS: OPER767 TERM PROJECT -

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 41	0.0000 =	0.0000	Row 42	0.0000 =	0.0000
Row 43	0.0000 =	0.0000	Row 44	0.0000 =	0.0000
Row 45	0.0000 =	0.0000	Row 46	0.0000 =	0.0000
Row 47	0.0000 =	0.0000	Row 48	0.0000 =	0.0000
Row 49	0.0000 =	0.0000	Row 50	0.0000 =	0.0000
Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000

Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000
Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

CONSTRAINTS: OPER767 TERM PROJECT -

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000
Row 67	0.0000 =	0.0000	Row 68	0.0000 =	0.0000
Row 69	0.0000 =	0.0000	Row 70	0.0000 =	0.0000
Row 71	0.0000 =	0.0000	Row 72	0.0000 =	0.0000
Row 73	0.0000 =	0.0000	Row 74	0.0000 =	0.0000
Row 75	0.0000 =	0.0000			

Total Error: 0.000000

Appendix E: MICROSOLVE (p=1)

ARC PARAMETERS AND FLOWS
SOLUTION COST = -69.99609

ARCS THAT START AT 1

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
10	1	0	1	-8	1	1
11	2	0	1	-5	1	0
12	3	0	1	-4	1	0
4	4	0	1	0	4	.25

ARCS THAT START AT 2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
10	5	0	1	-3	1	0
11	6	0	1	-6	1	0
12	7	0	1	-9	1	1
5	8	0	1	0	4	.4167

ARCS THAT START AT 3

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
10	9	0	1	-2	1	0
11	10	0	1	-7	1	1
12	11	0	1	-5	1	0
6	12	0	1	0	4	.3333

ARCS THAT START AT 4

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
13	13	0	1	-6	1	0
14	14	0	1	-2	1	0
15	15	0	1	-7	1	1
7	16	0	1	0	3	0

ARCS THAT START AT 5

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
13	17	0	1	-9	1	1
14	18	0	1	-3	1	0
15	19	0	1	-6	1	0
8	20	0	1	0	3	.6667

ARCS THAT START AT 6

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	21	0	1	-4	1	0
14	22	0	1	-6	1	1
15	23	0	1	-5	1	0
9	24	0	1	0	3	.3333

ARCS THAT START AT 7

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	25	0	1	-4	1	0
17	26	0	1	-8	1	0
18	27	0	1	-9	1	0

ARCS THAT START AT 8

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	28	0	1	-6	1	1
17	29	0	1	-7	1	0
18	30	0	1	-9	1	1

ARCS THAT START AT 9

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	31	0	1	-3	1	0
17	32	0	1	-9	1	1
18	33	0	1	-7	1	0

NO ARCS START AT NODE 10

NO ARCS START AT NODE 11

NO ARCS START AT NODE 12

NO ARCS START AT NODE 13

NO ARCS START AT NODE 14

NO ARCS START AT NODE 15

NO ARCS START AT NODE 16

NO ARCS START AT NODE 17

NO ARCS START AT NODE 18

ARCS THAT START AT 19

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
1	34	0	1	0	4	.3125
2	35	0	1	0	4	.3541667
3	36	0	1	0	4	.3333333
SLACK	37	0	1	99999	1	3.91E-08

ARC PARAMETERS AND FLOWS
SOLUTION COST = -119.9682

ARCS THAT START AT 1

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	1	0	1	-8	1	1
11	2	0	1	-5	1	0
12	3	0	1	-4	1	0
4	4	0	1	0	4	.75

ARCS THAT START AT 2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	5	0	1	-3	1	.9999997
11	6	0	1	-6	1	1
12	7	0	1	-9	1	1
5	8	0	1	0	4	.91667

ARCS THAT START AT 3

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	9	0	1	-2	1	0
11	10	0	1	-7	1	1
12	11	0	1	-5	1	1
6	12	0	1	0	4	.333334

ARCS THAT START AT 4

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	13	0	1	-6	1	1
14	14	0	1	-2	1	0
15	15	0	1	-7	1	1
7	16	0	1	0	3	1

ARCS THAT START AT 5

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	17	0	1	-9	1	1
14	18	0	1	-3	1	1
15	19	0	1	-6	1	1
8	20	0	1	0	3	.666667

ARCS THAT START AT 6

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
13	21	0	1	-4	1	0
14	22	0	1	-6	1	1
15	23	0	1	-5	1	0
9	24	0	1	0	3	.66667

ARCS THAT START AT 7

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
16	25	0	1	-4	1	.99999
17	26	0	1	-8	1	1
18	27	0	1	-9	1	1

ARCS THAT START AT 8

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
16	28	0	1	-6	1	1
17	29	0	1	-7	1	0
18	30	0	1	-9	1	1

ARCS THAT START AT 9

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
16	31	0	1	-3	1	0
17	32	0	1	-9	1	1
18	33	0	1	-7	1	0

NO ARCS START AT NODE 10

NO ARCS START AT NODE 11

NO ARCS START AT NODE 12

NO ARCS START AT NODE 13

NO ARCS START AT NODE 14

NO ARCS START AT NODE 15

NO ARCS START AT NODE 16

NO ARCS START AT NODE 17

NO ARCS START AT NODE 18

ARCS THAT START AT 19

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
1	34	0	1	0	4	.4375
2	35	0	1	0	4	.9791666
3	36	0	1	0	4	.5833333
SLACK	37	0	1	99999	1	0

Appendix G: Set Covering MIP83 (d=2)

..TITLE

SET COVERING MIP83 FORMULATION (d=2)

..OBJECTIVE MINIMIZE

*source connectors

1000 [[XS1] + 1000 [[XS2] + 1000 [[XS3]]

*state 1

+ 0 [[X110]] + 0 [[X111]] + 0 [[X112]]

+ 0 [[X210]] + 0 [[X211]] + 0 [[X212]]

+ 0 [[X310]] + 0 [[X311]] + 0 [[X312]]

*state 2

+ 0 [[X413]] + 0 [[X414]] + 0 [[X415]]

+ 0 [[X513]] + 0 [[X514]] + 0 [[X515]]

+ 0 [[X613]] + 0 [[X614]] + 0 [[X615]]

*state 3

+ 0 [[X716]] + 0 [[X717]] + 0 [[X718]]

+ 0 [[X816]] + 0 [[X817]] + 0 [[X818]]

+ 0 [[X916]] + 0 [[X917]] + 0 [[X918]]

*interstate links

+ 0 [[X14]] + 0 [[X25]] + 0 [[X36]]

+ 0 [[X47]] + 0 [[X58]] + 0 [[X69]]

*demand-sink connectors

+ 0 [X10T] + 0 [X11T] + 0 [X12T]

+ 0 [X13T] + 0 [X14T] + 0 [X15T]

+ 0 [X16T] + 0 [X17T] + 0 [X18T]

*excess-sink connectors

- 1 [X10TE] - 1 [X11TE] - 1 [X12TE]

- 1 [X13TE] - 1 [X14TE] - 1 [X15TE]

- 1 [X16TE] - 1 [X17TE] - 1 [X18TE]

..BOUNDS

*demand for mission 1 (nodes 10, 13, 16)

X10T >= 2

X13T >= 2

X16T >= 2

*demand for mission 2 (nodes 11, 14, 17)

X11T >= 2

X14T >= 2

X17T >= 2

*demand for mission 3 (nodes 12, 15, 18)

X12T >= 2

X15T >= 2

X18T >= 2

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 >= 1

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T + X10TE - 8 X110 - 3 X210 - 2 X310 = 0

NODE 11: X11T + X11TE - 5 X111 - 6 X211 - 7 X311 = 0

NODE 12: X12T + X12TE - 4 X112 - 9 X212 - 5 X312 = 0

NODE 13: X13T + X13TE - 6 X413 - 9 X513 - 4 X613 = 0

NODE 14: X14T + X14TE - 2 X414 - 3 X514 - 6 X614 = 0

NODE 15: X15T + X15TE - 7 X415 - 6 X515 - 5 X615 = 0

NODE 16: X16T + X16TE - 4 X716 - 6 X816 - 3 X916 = 0

NODE 17: X17T + X17TE - 8 X717 - 7 X817 - 9 X917 = 0

NODE 18: X18T + X18TE - 9 X718 - 9 X818 - 7 X918 = 0

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T <= 0

NODE TE: - X10TE - X11TE - X12TE
- X13TE - X14TE - X15TE
- X16TE - X17TE - X18TE <= 0

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MINIMIZED.
MIP Strategy: 1
Variables: 54
Integer: 54
Constraints: 21
2 LE, 18 EQ, 1 GE.
Non-zero LP elements: 108
Disk Space: 0K bytes.
Page Space: 9K bytes.
Capacity: 8.5% used.
Estimated Time: 00:00:15

Iter 46
Solution Time: 00:00:01

UNIQUE SOLUTION

Optimal Solution: 899.5714 Max Node Depth: 580 Limit: NONE

Solution: 960.0000 Iter: 71 Nodes: 2 Iteration Time: 00:00:11
INTEGER SOLUTION

File: SETCOV 8/25/90 16:43:02 Page 1-1
SOLUTION (Minimized): 960.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
XS1	0.0000	1,000.0000	XS2	1.0000	1,000.0000
XS3	0.0000	1,000.0000	X110	0.0000	0.0000
X111	0.0000	0.0000	X112	0.0000	0.0000
X210	1.0000	0.0000	X211	1.0000	0.0000
X212	1.0000	0.0000	X310	0.0000	0.0000
X311	0.0000	0.0000	X312	0.0000	0.0000
X413	0.0000	0.0000	X414	0.0000	0.0000
X415	0.0000	0.0000	X513	1.0000	0.0000
X514	1.0000	0.0000	X515	1.0000	0.0000
X613	0.0000	0.0000	X614	0.0000	0.0000

File: SETCOV 8/25/90 16:43:02 Page 1-2
SOLUTION (Minimized): 960.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.0000	0.0000	X716	0.0000	0.0000
X717	0.0000	0.0000	X718	0.0000	0.0000
X816	1.0000	0.0000	X817	1.0000	0.0000
X818	1.0000	0.0000	X916	0.0000	0.0000
X917	0.0000	0.0000	X918	0.0000	0.0000
X14	0.0000	0.0000	X25	1.0000	0.0000
X36	0.0000	0.0000	X47	0.0000	0.0000
X58	1.0000	0.0000	X69	0.0000	0.0000
X10T	2.0000	0.0000	X11T	2.0000	0.0000
X12T	2.0000	0.0000	X13T	2.0000	0.0000

File: SETCOV 8/25/90 16:43:02 Page 1-3

SOLUTION (Minimized): 960.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X14T	2.0000	0.0000	X15T	2.0000	0.0000
X16T	2.0000	0.0000	X17T	2.0000	0.0000
X18T	2.0000	0.0000	X10TE	1.0000	-1.0000
X11TE	4.0000	-1.0000	X12TE	7.0000	-1.0000
X13TE	7.0000	-1.0000	X14TE	1.0000	-1.0000
X15TE	4.0000	-1.0000	X16TE	4.0000	-1.0000
X17TE	5.0000	-1.0000	X18TE	7.0000	-1.0000

File: SETCOV 8/25/90 16:43:02 Page 1-4

CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	1.0000 >	1.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	-18.0000 <	0.0000

File: SETCOV 8/25/90 16:43:02 Page 1-5

CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE TE	-40.0000 <	0.0000			

Total Error: 0.000000

```
..TITLE
SET COVERING MIP83 FORMULATION (d=4)
..OBJECTIVE MINIMIZE
 *source connectors
  1000 [[XS1] + 1000 [[XS2] + 1000 [[XS3]]
 *state 1
 + 0 [[X110]] + 0 [[X111]] + 0 [[X112]]
 + 0 [[X210]] + 0 [[X211]] + 0 [[X212]]
 + 0 [[X310]] + 0 [[X311]] + 0 [[X312]]
 *state 2
 + 0 [[X413]] + 0 [[X414]] + 0 [[X415]]
 + 0 [[X513]] + 0 [[X514]] + 0 [[X515]]
 + 0 [[X613]] + 0 [[X614]] + 0 [[X615]]
 *state 3
 + 0 [[X716]] + 0 [[X717]] + 0 [[X718]]
 + 0 [[X816]] + 0 [[X817]] + 0 [[X818]]
 + 0 [[X916]] + 0 [[X917]] + 0 [[X918]]
 *interstate links
 + 0 [[X14]] + 0 [[X25]] + 0 [[X36]]
 + 0 [[X47]] + 0 [[X58]] + 0 [[X69]]
 *demand-sink connectors
 + 0 [X10T] + 0 [X11T] + 0 [X12T]
 + 0 [X13T] + 0 [X14T] + 0 [X15T]
 + 0 [X16T] + 0 [X17T] + 0 [X18T]
 *excess-sink connectors
```

- 1 [X10TE] - 1 [X11TE] - 1 [X12TE]

- 1 [X13TE] - 1 [X14TE] - 1 [X15TE]

- 1 [X16TE] - 1 [X17TE] - 1 [X18TE]

..BOUNDS

*demand for mission 1 (nodes 10, 13, 16)

X10T >= 4

X13T >= 4

X16T >= 4

*demand for mission 2 (nodes 11, 14, 17)

X11T >= 4

X14T >= 4

X17T >= 4

*demand for mission 3 (nodes 12, 15, 18)

X12T >= 4

X15T >= 4

X18T >= 4

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 >= 1

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T + X10TE - 8 X110 - 3 X210 - 2 X310 = 0

NODE 11: X11T + X11TE - 5 X111 - 6 X211 - 7 X311 = 0

NODE 12: X12T + X12TE - 4 X112 - 9 X212 - 5 X312 = 0

NODE 13: X13T + X13TE - 6 X413 - 9 X513 - 4 X613 = 0

NODE 14: X14T + X14TE - 2 X414 - 3 X514 - 6 X614 = 0

NODE 15: X15T + X15TE - 7 X415 - 6 X515 - 5 X615 = 0

NODE 16: X16T + X16TE - 4 X716 - 6 X816 - 3 X916 = 0

NODE 17: X17T + X17TE - 8 X717 - 7 X817 - 9 X917 = 0

NODE 18: X18T + X18TE - 9 X718 - 9 X818 - 7 X918 = 0

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T <= 0

NODE TE: - X10TE - X11TE - X12TE
- X13TE - X14TE - X15TE
- X16TE - X17TE - X18TE <= 0

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MINIMIZED.
MIP Strategy: 1
Variables: 54
Integer: 54
Constraints: 21
2 LE, 18 EQ, 1 GE.
Non-zero LP elements: 108
Disk Space: 0K bytes.
Page Space: 9K bytes.
Capacity: 8.5% used.
Estimated Time: 00:00:15

Iter 48
Solution Time: 00:00:02
UNIQUE SOLUTION

Optimal Solution: 919.2540 Max Node Depth: 580 Limit: NONE

Solution: 1,925.0000 Iter: 103 Nodes: 8 Iteration Time: 00:00:17
INTEGER SOLUTION

File: Setco4 9/04/90 00:05:48 Page 1-1
SOLUTION (Minimized): 1,925.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
XS1	1.0000	1,000.0000	XS2	1.0000	1,000.0000
XS3	0.0000	1,000.0000	X110	1.0000	0.0000
X111	1.0000	0.0000	X112	1.0000	0.0000
X210	1.0000	0.0000	X211	1.0000	0.0000
X212	1.0000	0.0000	X310	0.0000	0.0000
X311	0.0000	0.0000	X312	0.0000	0.0000
X413	1.0000	0.0000	X414	1.0000	0.0000
X415	1.0000	0.0000	X513	1.0000	0.0000
X514	1.0000	0.0000	X515	1.0000	0.0000
X613	0.0000	0.0000	X614	0.0000	0.0000

File: Setco4 9/04/90 00:05:48 Page 1-2
SOLUTION (Minimized): 1,925.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.0000	0.0000	X716	1.0000	0.0000
X717	1.0000	0.0000	X718	1.0000	0.0000
X816	1.0000	0.0000	X817	1.0000	0.0000
X818	1.0000	0.0000	X916	0.0000	0.0000
X917	0.0000	0.0000	X918	0.0000	0.0000
X14	1.0000	0.0000	X25	1.0000	0.0000
X36	0.0000	0.0000	X47	1.0000	0.0000
X58	1.0000	0.0000	X69	0.0000	0.0000
X10T	4.0000	0.0000	X11T	4.0000	0.0000

X12T	4.0000	0.0000	X13T	4.0000	0.0000

File: Setco4 9/04/90 00:05:48 Page 1-3
SOLUTION (Minimized): 1,925.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X14T	4.0000	0.0000	X15T	4.0000	0.0000
X16T	4.0000	0.0000	X17T	4.0000	0.0000
X18T	4.0000	0.0000	X10TE	7.0000	-1.0000
X11TE	7.0000	-1.0000	X12TE	9.0000	-1.0000
X13TE	11.0000	-1.0000	X14TE	1.0000	-1.0000
X15TE	9.0000	-1.0000	X16TE	6.0000	-1.0000
X17TE	11.0000	-1.0000	X18TE	14.0000	-1.0000

File: Setco4 9/04/90 00:05:48 Page 1-4
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	2.0000 >	1.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	-36.0000 <	0.0000

File: Setco4 9/04/90 00:05:48 Page 1-5
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE TE	-75.0000 <	0.0000			

Total Error: 0.000000

```
..TITLE
          SET COVERING MIP83 FORMULATION (d=6)
..OBJECTIVE MINIMIZE
*source connectors
 1000 [[XS1] + 1000 [[XS2] + 1000 [[XS3]]
*state 1
+ 0 [[X110]] + 0 [[X111]] + 0 [[X112]]
+ 0 [[X210]] + 0 [[X211]] + 0 [[X212]]
+ 0 [[X310]] + 0 [[X311]] + 0 [[X312]]
*state 2
+ 0 [[X413]] + 0 [[X414]] + 0 [[X415]]
+ 0 [[X513]] + 0 [[X514]] + 0 [[X515]]
+ 0 [[X613]] + 0 [[X614]] + 0 [[X615]]
*state 3
+ 0 [[X716]] + 0 [[X717]] + 0 [[X718]]
+ 0 [[X816]] + 0 [[X817]] + 0 [[X818]]
+ 0 [[X916]] + 0 [[X917]] + 0 [[X918]]
*interstate links
+ 0 [[X14]] + 0 [[X25]] + 0 [[X36]]
+ 0 [[X47]] + 0 [[X58]] + 0 [[X69]]
*demand-sink connectors
+ 0 [X10T] + 0 [X11T] + 0 [X12T]
+ 0 [X13T] + 0 [X14T] + 0 [X15T]
+ 0 [X16T] + 0 [X17T] + 0 [X18T]
```

*excess-sink connectors

- 1 [X10TE] - 1 [X11TE] - 1 [X12TE]

- 1 [X13TE] - 1 [X14TE] - 1 [X15TE]

- 1 [X16TE] - 1 [X17TE] - 1 [X18TE]

..BOUNDS

*demand for mission 1 (nodes 10, 13, 16)

X10T >= 6

X13T >= 6

X16T >= 6

*demand for mission 2 (nodes 11, 14, 17)

X11T >= 6

X14T >= 6

X17T >= 6

*demand for mission 3 (nodes 12, 15, 18)

X12T >= 6

X15T >= 6

X18T >= 6

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 >= 1

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T + X10TE - 8 X110 - 3 X210 - 2 X310 = 0

NODE 11: X11T + X11TE - 5 X111 - 6 X211 - 7 X311 = 0

NODE 12: X12T + X12TE - 4 X112 - 9 X212 - 5 X312 = 0

NODE 13: X13T + X13TE - 6 X413 - 9 X513 - 4 X613 = 0

NODE 14: X14T + X14TE - 2 X414 - 3 X514 - 6 X614 = 0

NODE 15: X15T + X15TE - 7 X415 - 6 X515 - 5 X615 = 0

NODE 16: X16T + X16TE - 4 X716 - 6 X816 - 3 X916 = 0

NODE 17: X17T + X17TE - 8 X717 - 7 X817 - 9 X917 = 0

NODE 18: X18T + X18TE - 9 X718 - 9 X818 - 7 X918 = 0

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T <= 0

NODE TE: - X10TE - X11TE - X12TE
- X13TE - X14TE - X15TE
- X16TE - X17TE - X18TE <= 0

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MINIMIZED.
MIP Strategy: 1
Variables: 54
Integer: 54
Constraints: 21
2 LE, 18 EQ, 1 GE.
Non-zero LP elements: 108
Disk Space: 0K bytes.
Page Space: 9K bytes.
Capacity: 8.5% used.
Estimated Time: 00:00:15

Iter 39
Solution Time: 00:00:02
ALTERNATE SOLUTIONS

Optimal Solution: 946.5714 Max Node Depth: 580 Limit: NONE

Solution: 1,953.0000 Iter: 57 Nodes: 3 Iteration Time: 00:00:10
INTEGER SOLUTION

File: Setco6 9/04/90 00:07:12 Page 1-1
SOLUTION (Minimized): 1,953.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
XS1	1.0000	1,000.0000	XS2	0.0000	1,000.0000
XS3	1.0000	1,000.0000	X110	1.0000	0.0000
X111	1.0000	0.0000	X112	1.0000	0.0000
X210	0.0000	0.0000	X211	0.0000	0.0000
X212	0.0000	0.0000	X310	1.0000	0.0000
X311	1.0000	0.0000	X312	1.0000	0.0000
X413	1.0000	0.0000	X414	1.0000	0.0000
X415	1.0000	0.0000	X513	0.0000	0.0000
X514	0.0000	0.0000	X515	0.0000	0.0000
X613	1.0000	0.0000	X614	1.0000	0.0000

File: Setco6 9/04/90 00:07:12 Page 1-2
SOLUTION (Minimized): 1,953.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X615	1.0000	0.0000	X716	1.0000	0.0000
X717	1.0000	0.0000	X718	1.0000	0.0000
X816	0.0000	0.0000	X817	0.0000	0.0000
X818	0.0000	0.0000	X916	1.0000	0.0000
X917	1.0000	0.0000	X918	1.0000	0.0000
X14	1.0000	0.0000	X25	0.0000	0.0000
X36	1.0000	0.0000	X47	1.0000	0.0000
X58	0.0000	0.0000	X69	1.0000	0.0000
X10T	6.0000	0.0000	X11T	6.0000	0.0000

X12T	6.0000	0.0000	X13T	6.0000	0.0000

File: Setco6 9/04/90 00:07:12 Page 1-3
SOLUTION (Minimized): 1,953.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X14T	6.0000	0.0000	X15T	6.0000	0.0000
X16T	6.0000	0.0000	X17T	6.0000	0.0000
X18T	6.0000	0.0000	X10TE	4.0000	-1.0000
X11TE	6.0000	-1.0000	X12TE	3.0000	-1.0000
X13TE	4.0000	-1.0000	X14TE	2.0000	-1.0000
X15TE	6.0000	-1.0000	X16TE	1.0000	-1.0000
X17TE	11.0000	-1.0000	X18TE	10.0000	-1.0000

File: Setco6 9/04/90 00:07:12 Page 1-4
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	2.0000 >	1.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	-54.0000 <	0.0000

File: Setco6 **9/04/90 00:07:12 Page 1-5**

CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity		RHS	Constraint	Activity	RHS
NODE TE	-47.0000	<	0.0000			

Total Error: 0.000000

```
..TITLE
        SET COVERING LP83 FORMULATION (d=2)
..OBJECTIVE MINIMIZE
*source connectors
 1000 XS1 + 1000 XS2 + 1000 XS3
*state 1
+ 0 X110 + 0 X111 + 0 X112
+ 0 X210 + 0 X211 + 0 X212
+ 0 X310 + 0 X311 + 0 X312
*state 2
+ 0 X413 + 0 X414 + 0 X415
+ 0 X513 + 0 X514 + 0 X515
+ 0 X613 + 0 X614 + 0 X615
*state 3
+ 0 X716 + 0 X717 + 0 X718
+ 0 X816 + 0 X817 + 0 X818
+ 0 X916 + 0 X917 + 0 X918
*interstate links
+ 0 X14 + 0 X25 + 0 X36
+ 0 X47 + 0 X58 + 0 X69
*demand-sink connectors
+ 0 X10T + 0 X11T + 0 X12T
+ 0 X13T + 0 X14T + 0 X15T
+ 0 X16T + 0 X17T + 0 X18T
```

*excess-sink connectors

- 1 X10TE - 1 X11TE - 1 X12TE

- 1 X13TE - 1 X14TE - 1 X15TE

- 1 X16TE - 1 X17TE - 1 X18TE

..BOUNDS

*demand for mission 1 (nodes 10, 13, 16)

X10T >= 2

X13T >= 2

X16T >= 2

*demand for mission 2 (nodes 11, 14, 17)

X11T >= 2

X14T >= 2

X17T >= 2

*demand for mission 3 (nodes 12, 15, 18)

X12T >= 2

X15T >= 2

X18T >= 2

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 >= 1

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T + X10TE - 8 X110 - 3 X210 - 2 X310 = 0

NODE 11: X11T + X11TE - 5 X111 - 6 X211 - 7 X311 = 0

NODE 12: X12T + X12TE - 4 X112 - 9 X212 - 5 X312 = 0

NODE 13: X13T + X13TE - 6 X413 - 9 X513 - 4 X613 = 0

NODE 14: X14T + X14TE - 2 X414 - 3 X514 - 6 X614 = 0

NODE 15: X15T + X15TE - 7 X415 - 6 X515 - 5 X615 = 0

NODE 16: X16T + X16TE - 4 X716 - 6 X816 - 3 X916 = 0

NODE 17: X17T + X17TE - 8 X717 - 7 X817 - 9 X917 = 0

NODE 18: X18T + X18TE - 9 X718 - 9 X818 - 7 X918 = 0

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T <= 0

NODE TE: - X10TE - X11TE - X12TE
- X13TE - X14TE - X15TE
- X16TE - X17TE - X18TE <= 0

* source connector flows
XS1 <= 1
XS2 <= 1
XS3 <= 1

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1 = 0
X110 - X14 = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2 = 0
X210 - X25 = 0

X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3 = 0
X310 - X36 = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47 = 0
X14 - X47 = 0

X513 - X514 = 0
X513 - X515 = 0
X514 - X515 = 0
X513 - X58 = 0
X25 - X58 = 0

X613 - X614 = 0
X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
LP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MINIMIZED.
Variables: 54
Constraints: 66
5 LE, 60 EQ, 1 GE.

Non-zero LP elements: 195
Disk Space: 0K bytes.
Page Space: 29K bytes.
Capacity: 15.4% used.
Estimated Time: 00:00:33

Iter 48
Solution Time: 00:00:02
May have A L T E R N A T E S O L U T I O N

File: Setco2 9/03/90 23:55:51 Page 1-1
SOLUTION (Minimized): 960.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
XS1	0.0000	1,000.0000	XS2	1.0000	1,000.00[illegible]
XS3	0.0000	1,000.0000	X110	0.0000	0.0000
X111	0.0000	0.0000	X112	0.0000	0.0000
X210	1.0000	0.0000	X211	1.0000	0.0000
X212	1.0000	0.0000	X310	0.0000	0.0000
X311	0.0000	0.0000	X312	0.0000	0.0000
X413	0.0000	0.0000	X414	0.0000	0.0000
X415	0.0000	0.0000	X513	1.0000	0.0000
X514	1.0000	0.0000	X515	1.0000	0.0000
X613	0.0000	0.0000	X614	0.0000	0.0000

File: Setco2 9/03/90 23:55:51 Page 1-2
SOLUTION (Minimized): 960.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.0000	0.0000	X716	0.0000	0.0000
X717	0.0000	0.0000	X71[illegible]	0.0000	0.0000
X816	1.0000	0.0000	X817	1.0000	0.0000
X818	1.0000	0.0000	X916	0.0000	0.0000
X917	0.0000	0.0000	X918	0.0000	0.0000
X14	0.0000	0.0000	X25	1.0000	0.0000
X36	0.0000	0.0000	X47	0.0000	0.0000
X58	1.0000	0.0000	X69	0.0000	0.0000
X10T	2.0000	0.0000	X11T	2.0000	0.0000

X12T	2.0000	0.0000	X13T	2.0000	0.0000

File: Setco2 9/03/90 23:55:51 Page 1-3
SOLUTION (Minimized): 960.0000 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X14T	2.0000	0.0000	X15T	2.0000	0.0000
X16T	2.0000	0.0000	X17T	2.0000	0.0000
X18T	2.0000	0.0000	X10TE	1.0000	-1.0000
X11TE	4.0000	-1.0000	X12TE	7.0000	-1.0000
X13TE	7.0000	-1.0000	X14TE	1.0000	-1.0000
X15TE	4.0000	-1.0000	X16TE	4.0000	-1.0000
X17TE	5.0000	-1.0000	X18TE	7.0000	-1.0000

File: Setco2 9/03/90 23:55:51 Page 1-4
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	1.0000 >	1.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	-18.0000 <	0.0000

File: Setco2 9/03/90 23:55:51 Page 1-5
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE TE	-40.0000 <	0.0000	Row 22	0.0000 <	1.0000

Row 23	1.0000 <	1.0000	Row 24	0.0000 <	1.0000
Row 25	0.0000 =	0.0000	Row 26	0.0000 =	0.0000
Row 27	0.0000 =	0.0000	Row 28	0.0000 =	0.0000
Row 29	0.0000 =	0.0000	Row 30	0.0000 =	0.0000
Row 31	0.0000 =	0.0000	Row 32	0.0000 =	0.0000
Row 33	0.0000 =	0.0000	Row 34	0.0000 =	0.0000
Row 35	0.0000 =	0.0000	Row 36	0.0000 =	0.0000
Row 37	0.0000 =	0.0000	Row 38	0.0000 =	0.0000
Row 39	0.0000 =	0.0000	Row 40	0.0000 =	0.0000

File: Setco2 9/03/90 23:55:51 Page 1-6
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 41	0.0000 =	0.0000	Row 42	0.0000 =	0.0000
Row 43	0.0000 =	0.0000	Row 44	0.0000 =	0.0000
Row 45	0.0000 =	0.0000	Row 46	0.0000 =	0.0000
Row 47	0.0000 =	0.0000	Row 48	0.0000 =	0.0000
Row 49	0.0000 =	0.0000	Row 50	0.0000 =	0.0000
Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000
Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000
Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

File: Setco2 9/03/90 23:55:51 Page 1-7
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000

Total Error: 0.000000

..TITLE

SET COVERING LP83 FORMULATION (d=4)

..OBJECTIVE MINIMIZE

*source connectors

1000 XS1 + 1000 XS2 + 1000 XS3

*state 1

+ 0 X110 + 0 X111 + 0 X112

+ 0 X210 + 0 X211 + 0 X212

+ 0 X310 + 0 X311 + 0 X312

*state 2

+ 0 X413 + 0 X414 + 0 X415

+ 0 X513 + 0 X514 + 0 X515

+ 0 X613 + 0 X614 + 0 X615

*state 3

+ 0 X716 + 0 X717 + 0 X718

+ 0 X816 + 0 X817 + 0 X818

+ 0 X916 + 0 X917 + 0 X918

*interstate links

+ 0 X14 + 0 X25 + 0 X36

+ 0 X47 + 0 X58 + 0 X69

*demand-sink connectors

+ 0 X10T + 0 X11T + 0 X12T

+ 0 X13T + 0 X14T + 0 X15T

+ 0 X16T + 0 X17T + 0 X18T

*excess-sink connectors

-1 X10TE -1 X11TE -1 X12TE

-1 X13TE -1 X14TE -1 X15TE

-1 X16TE -1 X17TE -1 X18TE

..BOUNDS

*demand for mission 1 (nodes 10, 13, 16)

X10T >= 4

X13T >= 4

X16T >= 4

*demand for mission 2 (nodes 11, 14, 17)

X11T >= 4

X14T >= 4

X17T >= 4

*demand for mission 3 (nodes 12, 15, 18)

X12T >= 4

X15T >= 4

X18T >= 4

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 >= 1

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T + X10TE - 8 X110 - 3 X210 - 2 X310 = 0

NODE 11: X11T + X11TE - 5 X111 - 6 X211 - 7 X311 = 0

NODE 12: X12T + X12TE - 4 X112 - 9 X212 - 5 X312 = 0

NODE 13: X13T + X13TE - 6 X413 - 9 X513 - 4 X613 = 0

NODE 14: X14T + X14TE - 2 X414 - 3 X514 - 6 X614 = 0

NODE 15: X15T + X15TE - 7 X415 - 6 X515 - 5 X615 = 0

NODE 16: X16T + X16TE - 4 X716 - 6 X816 - 3 X916 = 0

NODE 17: X17T + X17TE - 8 X717 - 7 X817 - 9 X917 = 0

NODE 18: X18T + X18TE - 9 X718 - 9 X818 - 7 X918 = 0

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T <= 0

NODE TE: - X10TE - X11TE - X12TE
- X13TE - X14TE - X15TE
- X16TE - X17TE - X18TE <= 0

* source connector flows
XS1 <= 1
XS2 <= 1
XS3 <= 1

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1 = 0
X110 - X14 = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2 = 0
X210 - X25 = 0

X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3 = 0
X310 - X36 = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47 = 0
X14 - X47 = 0

X513 - X514 = 0
X513 - X515 = 0
X514 - X515 = 0
X513 - X58 = 0
X25 - X58 = 0

X613 - X614 = 0
X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
LP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MINIMIZED.
Variables: 54
Constraints: 66
5 LE, 60 EQ, 1 GE.
Non-zero LP elements: 195

Disk Space: 0K bytes.
Page Space: 29K bytes.
Capacity: 15.4% used.
Estimated Time: 00:00:33

Iter 46
Solution Time: 00:00:03
May have A L T E R N A T E S O L U T I O N

File: Setco4 9/03/90 23:57:03 Page 1-1
SOLUTION (Minimized): 983.7143 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
XS1	0.2857	1,000.0000	XS2	0.2857	1,000.0000
XS3	0.4286	1,000.0000	X110	0.2857	0.0000
X111	0.2857	0.0000	X112	0.2857	0.0000
X210	0.2857	0.0000	X211	0.2857	0.0000
X212	0.2857	0.0000	X310	0.4286	0.0000
X311	0.4286	0.0000	X312	0.4286	0.0000
X413	0.2857	0.0000	X414	0.2857	0.0000
X415	0.2857	0.0000	X513	0.2857	0.0000
X514	0.2857	0.0000	X515	0.2857	0.0000
X613	0.4286	0.0000	X614	0.4286	0.0000

File: Setco4 9/03/90 23:57:03 Page 1-2
SOLUTION (Minimized): 983.7143 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.4286	0.0000	X716	0.2857	0.0000
X717	0.2857	0.0000	X718	0.2857	0.0000
X816	0.2857	0.0000	X817	0.2857	0.0000
X818	0.2857	0.0000	X916	0.4286	0.0000
X917	0.4286	0.0000	X918	0.4286	0.0000
X14	0.2857	0.0000	X25	0.2857	0.0000
X36	0.4286	0.0000	X47	0.2857	0.0000
X58	0.2857	0.0000	X69	0.4286	0.0000
X10T	4.0000	0.0000	X11T	4.0000	0.0000

X12T	4.0000	0.0000	X13T	4.0000	0.0000

File: Setco4 9/03/90 23:57:03 Page 1-3
SOLUTION (Minimized): 983.7143 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X14T	4.0000	0.0000	X15T	4.0000	0.0000
X16T	4.0000	0.0000	X17T	4.0000	0.0000
X18T	4.0000	0.0000	X10TE	0.0000	-1.0000
X11TE	2.1429	-1.0000	X12TE	1.8571	-1.0000
X13TE	2.0000	-1.0000	X14TE	0.0000	-1.0000
X15TE	1.8571	-1.0000	X16TE	0.1429	-1.0000
X17TE	4.1429	-1.0000	X18TE	4.1429	-1.0000

File: Setco4 9/03/90 23:57:03 Page 1-4
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	1.0000 >	1.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	-36.0000 <	0.0000

File: Setco4 9/03/90 23:57:03 Page 1-5
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE TE	-16.2857 <	0.0000	Row 22	0.2857 <	1.0000
Row 23	0.2857 <	1.0000	Row 24	0.4286 <	1.0000

Row 25	0.0000 =	0.0000	Row 26	0.0000 =	0.0000
Row 27	0.0000 =	0.0000	Row 28	0.0000 =	0.0000
Row 29	0.0000 =	0.0000	Row 30	0.0000 =	0.0000
Row 31	0.0000 =	0.0000	Row 32	0.0000 =	0.0000
Row 33	0.0000 =	0.0000	Row 34	0.0000 =	0.0000
Row 35	0.0000 =	0.0000	Row 36	0.0000 =	0.0000
Row 37	0.0000 =	0.0000	Row 38	0.0000 =	0.0000
Row 39	0.0000 =	0.0000	Row 40	0.0000 =	0.0000

File: Setco4 9/03/90 23:57:03 Page 1-6
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 41	0.0000 =	0.0000	Row 42	0.0000 =	0.0000
Row 43	0.0000 =	0.0000	Row 44	0.0000 =	0.0000
Row 45	0.0000 =	0.0000	Row 46	0.0000 =	0.0000
Row 47	0.0000 =	0.0000	Row 48	0.0000 =	0.0000
Row 49	0.0000 =	0.0000	Row 50	0.0000 =	0.0000
Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000
Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000
Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

File: Setco4 9/03/90 23:57:03 Page 1-7
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000

Total Error: 0.000000

```
..TITLE
        SET COVERING LP83 FORMULATION (d=6)
..OBJECTIVE MINIMIZE
*source connectors
 1000 XS1 + 1000 XS2 + 1000 XS3
*state 1
+ 0 X110 + 0 X111 + 0 X112
+ 0 X210 + 0 X211 + 0 X212
+ 0 X310 + 0 X311 + 0 X312
*state 2
+ 0 X413 + 0 X414 + 0 X415
+ 0 X513 + 0 X514 + 0 X515
+ 0 X613 + 0 X614 + 0 X615
*state 3
+ 0 X716 + 0 X717 + 0 X718
+ 0 X816 + 0 X817 + 0 X818
+ 0 X916 + 0 X917 + 0 X918
*interstate links
+ 0 X14 + 0 X25 + 0 X36
+ 0 X47 + 0 X58 + 0 X69
*demand-sink connectors
+ 0  X10T  + 0  X11T  + 0  X12T
+ 0  X13T  + 0  X14T  + 0  X15T
+ 0  X16T  + 0  X17T  + 0  X18T
```

*excess-sink connectors

- 1 X10TE - 1 X11TE - 1 X12TE

- 1 X13TE - 1 X14TE - 1 X15TE

- 1 X16TE - 1 X17TE - 1 X18TE

..BOUNDS

*demand for mission 1 (nodes 10, 13, 16)

X10T >= 6

X13T >= 6

X16T >= 6

*demand for mission 2 (nodes 11, 14, 17)

X11T >= 6

X14T >= 6

X17T >= 6

*demand for mission 3 (nodes 12, 15, 18)

X12T >= 6

X15T >= 6

X18T >= 6

..CONSTRAINTS

NODE S: XS1 + XS2 + XS3 >= 1

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T + X10TE - 8 X110 - 3 X210 - 2 X310 = 0

NODE 11: X11T + X11TE - 5 X111 - 6 X211 - 7 X311 = 0

NODE 12: X12T + X12TE - 4 X112 - 9 X212 - 5 X312 = 0

NODE 13: X13T + X13TE - 6 X413 - 9 X513 - 4 X613 = 0

NODE 14: X14T + X14TE - 2 X414 - 3 X514 - 6 X614 = 0

NODE 15: X15T + X15TE - 7 X415 - 6 X515 - 5 X615 = 0

NODE 16: X16T + X16TE - 4 X716 - 6 X816 - 3 X916 = 0

NODE 17: X17T + X17TE - 8 X717 - 7 X817 - 9 X917 = 0

NODE 18: X18T + X18TE - 9 X718 - 9 X818 - 7 X918 = 0

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T <= 0

NODE TE: - X10TE - X11TE - X12TE
- X13TE - X14TE - X15TE
- X16TE - X17TE - X18TE <= 0

* source connector flows
XS1 <= 1
XS2 <= 1
XS3 <= 1

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1 = 0
X110 - X14 = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2 = 0
X210 - X25 = 0

X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3 = 0
X310 - X36 = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47 = 0
X14 - X47 = 0

X513 - X514 = 0
X513 - X515 = 0
X514 - X515 = 0
X513 - X58 = 0
X25 - X58 = 0

X613 - X614 = 0
X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
LP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MINIMIZED.
Variables: 54
Constraints: 66
5 LE, 60 EQ, 1 GE.
Non-zero LP elements: 195

Disk Space: 0K bytes.
Page Space: 29K bytes.
Capacity: 15.4% used.
Estimated Time: 00:00:33

Iter 44
Solution Time: 00:00:02
May have A L T E R N A T E S O L U T I O N

File: Setco6 9/03/90 23:57:57 Page 1-1
SOLUTION (Minimized): 1,458.4444 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
XS1	0.4444	1,000.0000	XS2	0.3704	1,000.0000
XS3	0.6667	1,000.0000	X110	0.4444	0.0000
X111	0.4444	0.0000	X112	0.4444	0.0000
X210	0.3704	0.0000	X211	0.3704	0.0000
X212	0.3704	0.0000	X310	0.6667	0.0000
X311	0.6667	0.0000	X312	0.6667	0.0000
X413	0.4444	0.0000	X414	0.4444	0.0000
X415	0.4444	0.0000	X513	0.3704	0.0000
X514	0.3704	0.0000	X515	0.3704	0.0000
X613	0.6667	0.0000	X614	0.6667	0.0000

File: Setco6 9/03/90 23:57:57 Page 1-2
SOLUTION (Minimized): 1,458.4444 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X615	0.6667	0.0000	X716	0.4444	0.0000
X717	0.4444	0.0000	X718	0.4444	0.0000
X816	0.3704	0.0000	X817	0.3704	0.0000
X818	0.3704	0.0000	X916	0.6667	0.0000
X917	0.6667	0.0000	X918	0.6667	0.0000
X14	0.4444	0.0000	X25	0.3704	0.0000
X36	0.6667	0.0000	X47	0.4444	0.0000
X58	0.3704	0.0000	X69	0.6667	0.0000
X10T	6.0000	0.0000	X11T	6.0000	0.0000

X12T	6.0000	0.0000	X13T	6.0000	0.0000

File: Setco6 9/03/90 23:57:57 Page 1-3
SOLUTION (Minimized): 1,458.4444 OPER767 TERM PROJECT - SET COVERING FORMUL

Variable	Activity	Cost	Variable	Activity	Cost
X14T	6.0000	0.0000	X15T	6.0000	0.0000
X16T	6.0000	0.0000	X17T	6.0000	0.0000
X18T	6.0000	0.0000	X10TE	0.0000	-1.0000
X11TE	3.1111	-1.0000	X12TE	2.4444	-1.0000
X13TE	2.6667	-1.0000	X14TE	0.0000	-1.0000
X15TE	2.6667	-1.0000	X16TE	0.0000	-1.0000
X17TE	6.1481	-1.0000	X18TE	6.0000	-1.0000

File: Setco6 9/03/90 23:57:57 Page 1-4
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE S	1.4815 >	1.0000	NODE 1	0.0000 =	0.0000
NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000
NODE 18	0.0000 =	0.0000	NODE T	-54.0000 <	0.0000

File: Setco6 9/03/90 23:57:57 Page 1-5
CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE TE	-23.0370 <	0.0000	Row 22	0.4444 <	1.0000
Row 23	0.3704 <	1.0000	Row 24	0.6667 <	1.0000

Row 25	0.0000 =	0.0000	Row 26	0.0000 =	0.0000
Row 27	0.0000 =	0.0000	Row 28	0.0000 =	0.0000
Row 29	0.0000 =	0.0000	Row 30	0.0000 =	0.0000
Row 31	0.0000 =	0.0000	Row 32	0.0000 =	0.0000
Row 33	0.0000 =	0.0000	Row 34	0.0000 =	0.0000
Row 35	0.0000 =	0.0000	Row 36	0.0000 =	0.0000
Row 37	0.0000 =	0.0000	Row 38	0.0000 =	0.0000
Row 39	0.0000 =	0.0000	Row 40	0.0000 =	0.0000

File: Setco6 9/03/90 23:57:57 Page 1-6

CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 41	0.0000 =	0.0000	Row 42	0.0000 =	0.0000
Row 43	0.0000 =	0.0000	Row 44	0.0000 =	0.0000
Row 45	0.0000 =	0.0000	Row 46	0.0000 =	0.0000
Row 47	0.0000 =	0.0000	Row 48	0.0000 =	0.0000
Row 49	0.0000 =	0.0000	Row 50	0.0000 =	0.0000
Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000
Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000
Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

File: Setco6 9/03/90 23:57:57 Page 1-7

CONSTRAINTS: OPER767 TERM PROJECT - SET COVERING FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000

Total Error: 0.000000

ARC PARAMETERS AND FLOWS
SOLUTION COST = 258.2672

ARCS THAT START AT 1

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	1	0	1	0	8	.25
11	2	0	1	0	5	-2.98E-08
12	3	0	1	0	4	0
4	4	0	1	0	4	7.14E-02

ARCS THAT START AT 2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	5	0	1	0	3	0
11	6	0	1	0	6	0
12	7	0	1	0	9	.22222
5	8	0	1	0	4	.10185

ARCS THAT START AT 3

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	9	0	1	0	2	0
11	10	0	1	0	7	.2857143
12	11	0	1	0	5	0
6	12	0	1	0	4	.1018519

ARCS THAT START AT 4

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	13	0	1	0	6	0
14	14	0	1	0	2	0
15	15	0	1	0	7	.2857143
7	16	0	1	0	3	0

ARCS THAT START AT 5

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	17	0	1	0	9	.222222
14	18	0	1	0	3	0
15	19	0	1	0	6	0
8	20	0	1	0	3	.1851852

ARCS THAT START AT 6

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	21	0	1	0	4	0
14	22	0	1	0	6	.33333
15	23	0	1	0	5	0
9	24	0	1	0	3	7.40E-02

ARCS THAT START AT 7

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	25	0	1	0	4	0
17	26	0	1	0	8	0
18	27	0	1	0	9	0

ARCS THAT START AT 8

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	28	0	1	0	6	.33333
17	29	0	1	0	7	0
18	30	0	1	0	9	.22222

ARCS THAT START AT 9

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	31	0	1	0	3	0
17	32	0	1	0	9	.22222
18	33	0	1	0	7	0

ARCS THAT START AT 10

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	34	2	100001	0	1	2
21	35	0	99999	-1	1	0

ARCS THAT START AT 11

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	36	2	100001	0	1	2
21	37	0	99999	-1	1	0

ARCS THAT START AT 12

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	38	2	100001	0	1	2
21	39	0	99999	-1	1	0

ARCS THAT START AT 13

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	40	2	100001	0	1	2
21	41	0	99999	-1	1	0

ARCS THAT START AT 14

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	42	2	100001	0	1	2
21	43	0	99999	-1	1	0

ARCS THAT START AT 15

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	44	2	100001	0	1	2
21	45	0	99999	-1	1	0

ARCS THAT START AT 16

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	46	2	100001	0	1	2
21	47	0	99999	-1	1	0

ARCS THAT START AT 17

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	48	2	100001	0	1	2
21	49	0	99999	-1	1	0

ARCS THAT START AT 18

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	50	2	100001	0	1	2
21	51	0	99999	-1	1	0

ARCS THAT START AT 19

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
1	52	0	1	1000	4	8.03E-02
2	53	0	1	1000	4	8.10E-02
3	54	0	1	1000	4	9.68E-02

ARCS THAT START AT 20

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	55	0	99999	0	1	0

ARCS THAT START AT 21

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
SLACK	56	0	99999	0	1	0

ARC PARAMETERS AND FLOWS
SOLUTION COST = 516.5344

ARCS THAT START AT 1

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	1	0	1	0	8	.5
11	2	0	1	0	5	-5.96E-08
12	3	0	1	0	4	0
4	4	0	1	0	4	.1428571

ARCS THAT START AT 2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	5	0	1	0	3	0
11	6	0	1	0	6	0
12	7	0	1	0	9	.44444
5	8	0	1	0	4	.2037037

ARCS THAT START AT 3

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	9	0	1	0	2	0
11	10	0	1	0	7	.5714286
12	11	0	1	0	5	0
6	12	0	1	0	4	.2037037

ARCS THAT START AT 4

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	13	0	1	0	6	0
14	14	0	1	0	2	0
15	15	0	1	0	7	.5714286
7	16	0	1	0	3	0

ARCS THAT START AT 5

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	17	0	1	0	9	.444444
14	18	0	1	0	3	0
15	19	0	1	0	6	0
8	20	0	1	0	3	.3703704

ARCS THAT START AT 6

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
13	21	0	1	0	4	0
14	22	0	1	0	6	.666667
15	23	0	1	0	5	0
9	24	0	1	0	3	.1481481

ARCS THAT START AT 7

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
16	25	0	1	0	4	0
17	26	0	1	0	8	0
18	27	0	1	0	9	0

ARCS THAT START AT 8

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
16	28	0	1	0	6	.666667
17	29	0	1	0	7	0
18	30	0	1	0	9	.444444

ARCS THAT START AT 9

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
16	31	0	1	0	3	0
17	32	0	1	0	9	.444444
18	33	0	1	0	7	0

ARCS THAT START AT 10

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	34	2	100001	0	1	4
21	35	0	99999	-1	1	0

ARCS THAT START AT 11

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	36	2	100001	0	1	4
21	37	0	99999	-1	1	0

ARCS THAT START AT 12

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	38	2	100001	0	1	4
21	39	0	99999	-1	1	0

ARCS THAT START AT 13

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	40	2	100001	0	1	4
21	41	0	99999	-1	1	0

ARCS THAT START AT 14

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	42	2	100001	0	1	4
21	43	0	99999	-1	1	0

ARCS THAT START AT 15

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	44	2	100001	0	1	4
21	45	0	99999	-1	1	0

ARCS THAT START AT 16

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	46	2	100001	0	1	4
21	47	0	99999	-1	1	0

ARCS THAT START AT 17

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	48	2	100001	0	1	4
21	49	0	99999	-1	1	0

ARCS THAT START AT 18

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	50	2	100001	0	1	4
21	51	0	99999	-1	1	0

ARCS THAT START AT 19

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
1	52	0	1	1000	4	.1607143
2	53	0	1	1000	4	.162037
3	54	0	1	1000	4	.1937831

ARCS THAT START AT 20

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	55	0	99999	0	1	0

ARCS THAT START AT 21

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
SLACK	56	0	99999	0	1	0

ARC PARAMETERS AND FLOWS
SOLUTION COST = 774.8016

ARCS THAT START AT 1

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	1	0	1	0	8	.75
11	2	0	1	0	5	0
12	3	0	1	0	4	0
4	4	0	1	0	4	.2142857

ARCS THAT START AT 2

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	5	0	1	0	3	0
11	6	0	1	0	6	0
12	7	0	1	0	9	.66666
5	8	0	1	0	4	.30556

ARCS THAT START AT 3

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
10	9	0	1	0	2	0
11	10	0	1	0	7	.85714
12	11	0	1	0	5	0
6	12	0	1	0	4	.30556

ARCS THAT START AT 4

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	13	0	1	0	6	0
14	14	0	1	0	2	0
15	15	0	1	0	7	.85714
7	16	0	1	0	3	0

ARCS THAT START AT 5

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	17	0	1	0	9	.66667
14	18	0	1	0	3	0
15	19	0	1	0	6	0
8	20	0	1	0	3	.55556

ARCS THAT START AT 6

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
13	21	0	1	0	4	0
14	22	0	1	0	6	1
15	23	0	1	0	5	0
9	24	0	1	0	3	.22222

ARCS THAT START AT 7

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	25	0	1	0	4	0
17	26	0	1	0	8	0
18	27	0	1	0	9	0

ARCS THAT START AT 8

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	28	0	1	0	6	1
17	29	0	1	0	7	0
18	30	0	1	0	9	.66667

ARCS THAT START AT 9

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
16	31	0	1	0	3	0
17	32	0	1	0	9	.66667
18	33	0	1	0	7	0

ARCS THAT START AT 10

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	34	2	100001	0	1	6
21	35	0	99999	-1	1	0

ARCS THAT START AT 11

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	36	2	100001	0	1	6
21	37	0	99999	-1	1	0

ARCS THAT START AT 12

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	38	2	100001	0	1	6
21	39	0	99999	-1	1	0

ARCS THAT START AT 13

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	40	2	100001	0	1	6
21	41	0	99999	-1	1	0

ARCS THAT START AT 14

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	42	2	100001	0	1	6
21	43	0	99999	-1	1	0

ARCS THAT START AT 15

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	44	2	100001	0	1	6
21	45	0	99999	-1	1	0

ARCS THAT START AT 16

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	46	2	100001	0	1	6
21	47	0	99999	-1	1	0

ARCS THAT START AT 17

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	48	2	100001	0	1	6
21	49	0	99999	-1	1	0

ARCS THAT START AT 18

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
20	50	2	100001	0	1	6
21	51	0	99999	-1	1	0

ARCS THAT START AT 19

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
------	-------	------	------	-----	-----	-----
1	52	0	1	1000	4	.2410715
2	53	0	1	1000	4	.2430556
3	54	0	1	1000	4	.2906746

ARCS THAT START AT 20

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
20	55	0	99999	0	1	0

ARCS THAT START AT 21

GO TO	ARC NO.	LOWER	UPPER	COST	GAIN	FLOW
SLACK	56	0	99999	0	1	0

Appendix P: Max Coverage GAMS/BDMLP

```
GAMS 2.05 VAX VMS                    18-OCT-1990 11:43  PAGE     1
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
C O M P I L A T I O N


SETS
    I   LOCATIONS    /1*3/
    J   STATES       /1*3/
    K   SATELLITES   /1*3/;

TABLE
    W(I,J,K)  NUMBER OF OBSERVATION BLOCKS

    1   2   3

1.1 8   5   4
2.1 3   6   9
3.1 2   7   5

1.2 6   2   7
2.2 9   3   6
3.2 4   6   5

1.3 4   8   9
2.3 6   7   9
3.3 3   9   7;

PARAMETERS   CTRLI(I) SELECTION CONTROL /1 9,2 9,3 9/
         CTRL(J)  EQUATION CONTROL /1*2 1, 3 0/;

SCALARS
    M       NUMBER OF STATES              /3/
    S       NUMBER OF SATELLITES          /3/
    P       NUMBER OF FACILITIES SELECTED /2/
    GAIN    ARC GAIN                      /4/;

VARIABLES
    X(I,J,K)  BIPARTITE FLOW
    T(J,K)    SINK IN-FLOW
    Y(I,J)    SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW
    Z         OPTIMIZATION VARIABLE;

POSITIVE VARIABLES  X,T,Y;

    X.UP(I,J,K) = 1;
    Y.UP(I,J)   = 1;
    T.UP(J,K)   = P;

EQUATIONS
    OBJ          OBJECTIVE FUNCTION
```

```
    SRCE         SOURCE FLOW
    FNODE(I,J)     FACILITY NODE FLOW
    FNODEJ(I,J)   STATE J FACILITY NODE FLOW
    SNODE(J,K)    SATELLITE NODE FLOW
    SINK         SINK FLOW
    PAIR1(I,J,K)  EQUI DISTRIBUTION OF FLOW
    PAIR2(I)      EQUI DISTRIBUTION OF FLOW
    PAIR3(I,J)    EQUI DISTRIBUTION OF FLOW
    PAIR4(I,J)    EQUI DISTRIBUTION OF FLOW
```

```
GAMS 2.05 VAX VMS                    18-OCT-1990 11:43  PAGE      2
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
C O M P I L A T I O N
```

```

    SELECT1(I)    SELECT FACILITY
    SELECT0(I)    DO NOT SELECT FACILITY;

OBJ..
    Z =E= SUM((I,J,K),W(I,J,K) * X(I,J,K));

SRCE..
    SUM(I,Y(I,'1')) =E= P;

FNODE(I,J)$(CTRL(J) NE 0)..
    SUM(K,X(I,J,K)) + Y(I,J+1) - GAIN * Y(I,J) =E= 0;

FNODEJ(I,J)$(CTRL(J) EQ 0)..
    SUM(K,X(I,J,K)) - (GAIN - 1) * Y(I,J) =E= 0;

SNODE(J,K)..
    T(J,K) - SUM(I,X(I,J,K)) =E= 0;

SINK..
    SUM((J,K),T(J,K)) =E= M * S * P;

PAIR1(I,J,K)..
    X(I,J,'1') - X(I,J,K) =E= 0;

PAIR2(I)..
    X(I,'1','1') - Y(I,'1') =E= 0;

PAIR3(I,J)..
    X(I,J,'2') - X(I,J,'3') =E= 0;

PAIR4(I,J)$(CTRL(J) NE 0)..
    X(I,J,'1') - Y(I,J+1) =E= 0;

SELECT1(I)$(CTRLI(I) EQ 1)..
    Y(I,'1') =E= 1;
```

```
 91
 92 SELECT0(I)$(CTRLI(I) EQ 0)..
 93     Y(I,'1') =E= 0;
 94
 95
 96 MODEL MAXCOVER /ALL/;
 97 SOLVE MAXCOVER USING LP MAXIMIZING Z;
 98 DISPLAY X.L,Y.L,T.L;
 99
100


GAMS 2.05 VAX VMS                    18-OCT-1990 11:43  PAGE    3
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
SYMBOL LISTING


SYMBOL     TYPE   REFERENCES

CTRL      PARAM  DECLARED      24  DEFINED      24      REF      65
                 68      86
CTRLI     PARAM  DECLARED      23  DEFINED      23      REF      89
                 92
FNODE      EQU   DECLARED      47  DEFINED      66 IMPL-ASN      97
                REF      96
FNODEJ     EQU   DECLARED      48  DEFINED      69 IMPL-ASN      97
                REF      96
GAIN      PARAM  DECLARED      30  DEFINED      30      REF      66
                 69
I          SET   DECLARED       2  DEFINED       2      REF       7
                 23      33      35      47      48      51
                 52      53      54      56      57    2*60
                 63    3*66    2*69      72    2*78    2*81
               2*84    2*87      89      90      92      93
             CONTROL      40      41      60      63      65
                 68      72      77      80      83      86
                 89      92
J          SET   DECLARED       3  DEFINED       3      REF       7
                 24      33      34      35      47      48
                 49      51      53      54    2*60      65
               3*66      68    2*69    2*72      75    2*78
               2*84      86    2*87 CONTROL      40      41
                 42      60      65      68      71      75
                 77      83      86
K          SET   DECLARED       4  DEFINED       4      REF       7
                 33      34      49      51    2*60      66
                 69    2*72      75      78 CONTROL      40
                 42      60      66      69      71      75
                 77
M         PARAM  DECLARED      27  DEFINED      27      REF      75
MAXCOVER  MODEL  DECLARED      96  DEFINED      96      REF      97
OBJ        EQU   DECLARED      45  DEFINED      60 IMPL-ASN      97
```

```
            REF      96
P        PARAM DECLARED     29  DEFINED     29     REF      42
            63      75
PAIR1       EQU   DECLARED     51  DEFINED      78 IMPL-ASN      97
            REF      96
PAIR2       EQU   DECLARED     52  DEFINED      81 IMPL-ASN      97
            REF      96
PAIR3       EQU   DECLARED     53  DEFINED      84 IMPL-ASN      97
            REF      96
PAIR4       EQU   DECLARED     54  DEFINED      87 IMPL-ASN      97
            REF      96
S        PARAM DECLARED     28  DEFINED     28     REF      75
SELECT0     EQU   DECLARED     57  DEFINED      93 IMPL-ASN      97
            REF      96
SELECT1     EQU   DECLARED     56  DEFINED      90 IMPL-ASN      97
            REF      96
SINK        EQU   DECLARED     50  DEFINED      75 IMPL-ASN      97
            REF      96
SNODE       EQU   DECLARED     49  DEFINED      72 IMPL-ASN      97


GAMS 2.05 VAX VMS                 18-OCT-1990 11:43  PAGE     4
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
SYMBOL LISTING


SYMBOL    TYPE   REFERENCES

            REF      96
SRCE        EQU   DECLARED     46  DEFINED      63 IMPL-ASN      97
            REF      96
T        VAR   DECLARED     34 IMPL-ASN      97 ASSIGNED      42
            REF      38      72      75      98
W        PARAM DECLARED      7  DEFINED      7     REF      60
X        VAR   DECLARED     33 IMPL-ASN      97 ASSIGNED      40
            REF      38      60      66      69      72
           2*78      81    2*84      87      98
Y        VAR   DECLARED     35 IMPL-ASN      97 ASSIGNED      41
            REF      38      63    2*66      69      81
            87      90      93      93
Z        VAR   DECLARED     36 IMPL-ASN      97     REF      60
            97


SETS

I        LOCATIONS
J        STATES
K        SATELLITES


PARAMETERS
```

```
CTRL      EQUATION CONTROL
CTRLI     SELECTION CONTROL
GAIN      ARC GAIN
M         NUMBER OF STATES
P         NUMBER OF FACILITIES SELECTED
S         NUMBER OF SATELLITES
W         NUMBER OF OBSERVATION BLOCKS


VARIABLES

T         SINK IN-FLOW
X         BIPARTITE FLOW
Y         SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW
Z         OPTIMIZATION VARIABLE


EQUATIONS

FNODE     FACILITY NODE FLOW
FNODEJ    STATE J FACILITY NODE FLOW
OBJ       OBJECTIVE FUNCTION
PAIR1     EQUI DISTRIBUTION OF FLOW
PAIR2     EQUI DISTRIBUTION OF FLOW
PAIR3     EQUI DISTRIBUTION OF FLOW
PAIR4     EQUI DISTRIBUTION OF FLOW
SELECT0   DO NOT SELECT FACILITY


GAMS 2.05 VAX VMS                    18-OCT-1990 11:43  PAGE    5
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
SYMBOL LISTING


EQUATIONS

SELECT1   SELECT FACILITY
SINK      SINK FLOW
SNODE     SATELLITE NODE FLOW
SRCE      SOURCE FLOW


MODELS

MAXCOVER


COMPILATION TIME    =      0.590 SECONDS
```

```
---- OBJ      =E=  OBJECTIVE FUNCTION

OBJ..  - 8*X(1,1,1) - 5*X(1,1,2) - 4*X(1,1,3) - 6*X(1,2,1) - 2*X(1,2,2)

     - 7*X(1,2,3) - 4*X(1,3,1) - 8*X(1,3,2) - 9*X(1,3,3) - 3*X(2,1,1)

     - 6*X(2,1,2) - 9*X(2,1,3) - 9*X(2,2,1) - 3*X(2,2,2) - 6*X(2,2,3)

     - 6*X(2,3,1) - 7*X(2,3,2) - 9*X(2,3,3) - 2*X(3,1,1) - 7*X(3,1,2)

     - 5*X(3,1,3) - 4*X(3,2,1) - 6*X(3,2,2) - 5*X(3,2,3) - 3*X(3,3,1)

     - 9*X(3,3,2) - 7*X(3,3,3) + Z =E= 0 ;

---- SRCE     =E=  SOURCE FLOW

SRCE..  Y(1,1) + Y(2,1) + Y(3,1) =E= 2 ;

---- FNODE    =E=  FACILITY NODE FLOW

FNODE(1,1)..  X(1,1,1) + X(1,1,2) + X(1,1,3) - 4*Y(1,1) + Y(1,2) =E= 0 ;

FNODE(1,2)..  X(1,2,1) + X(1,2,2) + X(1,2,3) - 4*Y(1,2) + Y(1,3) =E= 0 ;

FNODE(2,1)..  X(2,1,1) + X(2,1,2) + X(2,1,3) - 4*Y(2,1) + Y(2,2) =E= 0 ;
REMAINING 3 ENTRIES SKIPPED

---- FNODEJ   =E=  STATE J FACILITY NODE FLOW

FNODEJ(1,3)..  X(1,3,1) + X(1,3,2) + X(1,3,3) - 3*Y(1,3) =E= 0 ;

FNODEJ(2,3)..  X(2,3,1) + X(2,3,2) + X(2,3,3) - 3*Y(2,3) =E= 0 ;

FNODEJ(3,3)..  X(3,3,1) + X(3,3,2) + X(3,3,3) - 3*Y(3,3) =E= 0 ;
```

EQUATION LISTING SOLVE MAXCOVER USING LP FROM LINE 97

```
---- SNODE     =E=  SATELLITE NODE FLOW

SNODE(1,1)..  - X(1,1,1) - X(2,1,1) - X(3,1,1) + T(1,1) =E= 0 ;

SNODE(1,2)..  - X(1,1,2) - X(2,1,2) - X(3,1,2) + T(1,2) =E= 0 ;

SNODE(1,3)..  - X(1,1,3) - X(2,1,3) - X(3,1,3) + T(1,3) =E= 0 ;
REMAINING 6 ENTRIES SKIPPED

---- SINK     =E=  SINK FLOW

SINK..  T(1,1) + T(1,2) + T(1,3) + T(2,1) + T(2,2) + T(2,3) + T(3,1) + T(3,2)

     + T(3,3) =E= 18 ;

---- PAIR1     =E=  EQUI DISTRIBUTION OF FLOW

PAIR1(1,1,2)..  X(1,1,1) - X(1,1,2) =E= 0 ;

PAIR1(1,1,3)..  X(1,1,1) - X(1,1,3) =E= 0 ;

PAIR1(1,2,2)..  X(1,2,1) - X(1,2,2) =E= 0 ;
REMAINING 15 ENTRIES SKIPPED

---- PAIR2     =E=  EQUI DISTRIBUTION OF FLOW

PAIR2(1)..  X(1,1,1) - Y(1,1) =E= 0 ;

PAIR2(2)..  X(2,1,1) - Y(2,1) =E= 0 ;

PAIR2(3)..  X(3,1,1) - Y(3,1) =E= 0 ;
```

---- PAIR3 =E= EQUI DISTRIBUTION OF FLOW

PAIR3(1,1).. X(1,1,2) - X(1,1,3) =E= 0 ;

PAIR3(1,2).. X(1,2,2) - X(1,2,3) =E= 0 ;

PAIR3(1,3).. X(1,3,2) - X(1,3,3) =E= 0 ;

REMAINING 6 ENTRIES SKIPPED

---- PAIR4 =E= EQUI DISTRIBUTION OF FLOW

PAIR4(1,1).. X(1,1,1) - Y(1,2) =E= 0 ;

PAIR4(1,2).. X(1,2,1) - Y(1,3) =E= 0 ;

PAIR4(2,1).. X(2,1,1) - Y(2,2) =E= 0 ;

REMAINING 3 ENTRIES SKIPPED

---- SELECT1 =E= SELECT FACILITY

NONE

---- SELECT0 =E= DO NOT SELECT FACILITY

NONE

GAMS 2.05 VAX VMS 18-OCT-1990 11:43 PAGE 9
G E N E R A L A L G E B R A I C M O D E L I N G S Y S T E M
COLUMN LISTING SOLVE MAXCOVER USING LP FROM LINE 97

```
---- X         BIPARTITE FLOW


X(1,1,1)
               (.LO, .L, .UP = 0, 0, 1)
      -8       OBJ
       1       FNODE(1,1)
      -1       SNODE(1,1)
       1       PAIR1(1,1,2)
       1       PAIR1(1,1,3)
       1       PAIR2(1)
       1       PAIR4(1,1)

X(1,1,2)
               (.LO, .L, .UP = 0, 0, 1)
      -5       OBJ
       1       FNODE(1,1)
      -1       SNODE(1,2)
      -1       PAIR1(1,1,2)
       1       PAIR3(1,1)

X(1,1,3)
               (.LO, .L, .UP = 0, 0, 1)
      -4       OBJ
       1       FNODE(1,1)
      -1       SNODE(1,3)
      -1       PAIR1(1,1,3)
      -1       PAIR3(1,1)

REMAINING 24 ENTRIES SKIPPED


---- T         SINK IN-FLOW


T(1,1)
               (.LO, .L, .UP = 0, 0, 2)
       1       SNODE(1,1)
       1       SINK

T(1,2)
               (.LO, .L, .UP = 0, 0, 2)
       1       SNODE(1,2)
       1       SINK

T(1,3)
               (.LO, .L, .UP = 0, 0, 2)
       1       SNODE(1,3)
       1       SINK

REMAINING 6 ENTRIES SKIPPED
```

COLUMN LISTING SOLVE MAXCOVER USING LP FROM LINE 97

```
---- Y        SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW


Y(1,1)
            (.LO, .L, .UP = 0, 0, 1)
      1     SRCE
     -4     FNODE(1,1)
     -1     PAIR2(1)

Y(1,2)
            (.LO, .L, .UP = 0, 0, 1)
      1     FNODE(1,1)
     -4     FNODE(1,2)
     -1     PAIR4(1,1)

Y(1,3)
            (.LO, .L, .UP = 0, 0, 1)
      1     FNODE(1,2)
     -3     FNODEJ(1,3)
     -1     PAIR4(1,2)

REMAINING 6 ENTRIES SKIPPED


---- Z        OPTIMIZATION VARIABLE


Z
            (.LO, .L, .UP =  -INF, 0,  +INF)
      1     OBJ
```


MODEL STATISTICS SOLVE MAXCOVER USING LP FROM LINE 97

```
MODEL STATISTICS

BLOCKS OF EQUATIONS   12     SINGLE EQUATIONS   57
BLOCKS OF VARIABLES    4     SINGLE VARIABLES   46
NON ZERO ELEMENTS    190

GENERATION TIME    =      0.930 SECONDS
```

```
EXECUTION TIME      =      1.550 SECONDS


GAMS 2.05 VAX VMS                    18-OCT-1990 11:44  PAGE    12
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
SOLUTION REPORT    SOLVE MAXCOVER USING LP FROM LINE 97


          S O L V E      S U M M A R Y

   MODEL   MAXCOVER          OBJECTIVE  Z
   TYPE    LP              DIRECTION  MAXIMIZE
   SOLVER  BDMLP              FROM LINE  97

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       1 OPTIMAL
**** OBJECTIVE VALUE             111.0000

 RESOURCE USAGE, LIMIT          0.530      1000.000
 ITERATION COUNT, LIMIT        31          1000

 BDM - LP  VERSION 1.01

 A. BROOKE, A. DRUD, AND A. MEERAUS,
 ANALYTIC SUPPORT UNIT,
 DEVELOPMENT RESEARCH DEPARTMENT,
 WORLD BANK,
 WASHINGTON, D.C. 20433, U.S.A.

 WORK SPACE NEEDED (ESTIMATE)  --    7107 WORDS.
 WORK SPACE AVAILABLE           --    7107 WORDS.

 EXIT -- OPTIMAL SOLUTION FOUND.

                  LOWER     LEVEL     UPPER    MARGINAL

---- EQU OBJ            .         .         .       1.000
---- EQU SRCE         2.000     2.000     2.000    -4.600

  OBJ       OBJECTIVE FUNCTION
  SRCE       SOURCE FLOW


---- EQU FNODE       FACILITY NODE FLOW

      LOWER     LEVEL     UPPER    MARGINAL

1.1      .         .         .       -2.400
1.2      .         .         .       -0.600
2.1      .         .         .       -1.150
2.2      .         .         .       -0.850
```

	LOWER	LEVEL	UPPER	MARGINAL
3.1	.	.	.	-1.150
3.2	.	.	.	0.150

---- EQU FNODEJ STATE J FACILITY NODE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.3	.	.	.	0.600
2.3	.	.	.	-0.733
3.3	.	.	.	-0.067

---- EQU SNODE SATELLITE NODE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	-6.400
1.2	.	.	.	-6.400
1.3	.	.	.	-6.400
2.1	.	.	.	-6.400
2.2	.	.	.	-6.400
2.3	.	.	.	-6.400
3.1	.	.	.	-6.400
3.2	.	.	.	-6.400
3.3	.	.	.	-6.400

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU SINK	18.000	18.000	18.000	6.400

SINK SINK FLOW

---- EQU PAIR1 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1.2	.	.	.	-1.000
1.1.3	.	.	.	.
1.2.2	.	.	.	2.600
1.2.3	.	.	.	.
1.3.2	.	.	.	-1.000
1.3.3	.	.	.	-2.000
2.1.2	.	.	.	.
2.1.3	.	.	.	-4.500
2.2.2	.	.	.	.

2.2.3	.	.	.	2.100
2.3.2	.	.	.	-4.667
2.3.3	.	.	.	.
3.1.2	.	.	.	.
3.1.3	.	.	.	-1.500
3.2.2	.	.	.	-2.900
3.2.3	.	.	.	.
3.3.2	.	.	.	-3.333
3.3.3	.	.	.	.

---- EQU PAIR2 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1	.	.	.	5.000
2	.	.	.	.
3	.	.	.	.

---- EQU PAIR3 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	.
1.2	.	.	.	-1.200
1.3	.	.	.	.
2.1	.	.	.	0.750
2.2	.	.	.	-2.550
2.3	.	.	.	-3.333
3.1	.	.	.	1.750
3.2	.	.	.	-3.450
3.3	.	.	.	-0.667

---- EQU PAIR4 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	.
1.2	.	.	.	-2.400
2.1	.	.	.	2.250
2.2	.	.	.	1.350
3.1	.	.	.	-1.750
3.2	.	.	.	0.350

---- EQU SELECT1 SELECT FACILITY

NONE

---- EQU SELECT0 DO NOT SELECT FACILITY

NONE

---- VAR X BIPARTITE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	1.000	1.000	.
1.1.2	.	1.000	1.000	.
1.1.3	.	1.000	1.000	.
1.2.1	.	1.000	1.000	.
1.2.2	.	1.000	1.000	.
1.2.3	.	1.000	1.000	.
1.3.1	.	1.000	1.000	.
1.3.2	.	1.000	1.000	.
1.3.3	.	1.000	1.000	.
2.1.1	.	1.000	1.000	.
2.1.2	.	1.000	1.000	.
2.1.3	.	1.000	1.000	.
2.2.1	.	1.000	1.000	.

VAR X BIPARTITE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
2.2.2	.	1.000	1.000	.
2.2.3	.	1.000	1.000	.
2.3.1	.	1.000	1.000	5.000
2.3.2	.	1.000	1.000	.
2.3.3	.	1.000	1.000	.
3.1.1	.	.	1.000	.
3.1.2	.	.	1.000	.
3.1.3	.	.	1.000	.
3.2.1	.	.	1.000	.
3.2.2	.	.	1.000	.
3.2.3	.	.	1.000	-5.000
3.3.1	.	.	1.000	.
3.3.2	.	.	1.000	.
3.3.3	.	.	1.000	.

---- VAR T SINK IN-FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	2.000	2.000	.
1.2	.	2.000	2.000	.
1.3	.	2.000	2.000	.
2.1	.	2.000	2.000	.
2.2	.	2.000	2.000	.
2.3	.	2.000	2.000	.
3.1	.	2.000	2.000	.
3.2	.	2.000	2.000	.
3.3	.	2.000	2.000	.

---- VAR Y SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	1.000	1.000	.
1.2	.	1.000	1.000	.
1.3	.	1.000	1.000	.
2.1	.	1.000	1.000	.
2.2	.	1.000	1.000	.
2.3	.	1.000	1.000	.
3.1	.	.	1.000	.
3.2	.	.	1.000	.
3.3	.	.	1.000	.

SOLUTION REPORT SOLVE MAXCOVER USING LP FROM LINE 97

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR Z	-INF	111.000	+INF	.

Z OPTIMIZATION VARIABLE

**** REPORT SUMMARY : 0 NONOPT
0 INFEASIBLE
0 UNBOUNDED

```
----    98 VARIABLE  X.L          BIPARTITE FLOW

            1          2          3

1.1     1.000      1.000      1.000
1.2     1.000      1.000      1.000
1.3     1.000      1.000      1.000
2.1     1.000      1.000      1.000
2.2     1.000      1.000      1.000
2.3     1.000      1.000      1.000


----    98 VARIABLE  Y.L          SOURCE OUT-FLOW AND INTERSTATE
CONNECTOR
                                  FLOW

          1          2          3

1     1.000      1.000      1.000
2     1.000      1.000      1.000


----    98 VARIABLE  T.L          SINK IN-FLOW

          1          2          3

1     2.000      2.000      2.000
2     2.000      2.000      2.000
3     2.000      2.000      2.000


**** FILE SUMMARY

INPUT   GSO91M:[PFORQUES]MAXCOVR2.GMS;5
OUTPUT  GSO91M:[PFORQUES]MAXCOVR2.LIS;5

EXECUTION TIME      =       1.010 SECONDS
```

```
GAMS 2.05 VAX VMS                    18-OCT-1990 11:18  PAGE     1
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
C O M P I L A T I O N


SETS
    I   LOCATIONS     /1*3/
    J   STATES        /1*3/
    K   SATELLITES    /1*3/;

TABLE
    A(I,J,K)  NUMBER OF OBSERVATION BLOCKS

     1   2   3

1.1  8   5   4
2.1  3   6   9
3.1  2   7   5

1.2  6   2   7
2.2  9   3   6
3.2  4   6   5

1.3  4   8   9
2.3  6   7   9
3.3  3   9   7;

PARAMETERS   CTRLI(I) SELECTION CONTROL /1 9,2 9,3 9/
             CTRLJ(J) EQUATION CONTROL /1*2 1, 3 0/
               D(K)  REQUIREMENTS VECTOR   /1*3 6/
               M(I)  COST VECTOR        /1*3 1000/
               W(K)  COST VECTOR          /1*3 -1/;

SCALAR
    GAIN    ARC GAIN    /4/;

VARIABLES
    X(I,J,K)  BIPARTITE FLOW
    T(J,K)    SINK T IN-FLOW
    TE(J,K)   SINK TE IN-FLOW
    Y(I,J)    SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW
    Z         OPTIMIZATION VARIABLE;

POSITIVE VARIABLES  X,T,TE,Y;
    X.UP(I,J,K) = 1;
    Y.UP(I,J)   = 1;

EQUATIONS
    OBJ         OBJECTIVE FUNCTION
```

```
    SRCE          SOURCE FLOW
    FNODE(I,J)    FACILITY NODE FLOW
    FNODEJ(I,J)   STATE J FACILITY NODE FLOW
    SNODE(J,K)    SATELLITE NODE FLOW
    PAIR1(I,J,K)  EQUI DISTRIBUTION OF FLOW
    PAIR2(I)      EQUI DISTRIBUTION OF FLOW
    PAIR3(I,J)    EQUI DISTRIBUTION OF FLOW
    PAIR4(I,J)    EQUI DISTRIBUTION OF FLOW
    REQ(J,K)      REQUIREMENTS
    SELECT1(I)    SELECT FACILITY
```

GAMS 2.05 VAX VMS 18-OCT-1990 11:18 PAGE 2
GENERAL ALGEBRAIC MODELING SYSTEM
COMPILATION

```
    SELECT0(I)    DO NOT SELECT FACILITY;


OBJ..
    Z =E= SUM(I,M(I) * Y(I,'1')) + SUM((J,K),W(K) * TE(J,K));

SRCE..
    SUM(I,Y(I,'1')) =G= 1;

FNODE(I,J)$(CTRLJ(J) NE 0)..
    SUM(K,X(I,J,K)) + Y(I,J+1) - GAIN*Y(I,J) =E= 0;

FNODEJ(I,J)$(CTRLJ(J) EQ 0)..
    SUM(K,X(I,J,K)) - (GAIN - 1)*Y(I,J) =E= 0;

SNODE(J,K)..
    T(J,K) + TE(J,K) - SUM(I,X(I,J,K) * A(I,J,K)) =E= 0;

PAIR1(I,J,K)..
    X(I,J,'1') - X(I,J,K) =E= 0;

PAIR2(I)..
    X(I,'1','1') - Y(I,'1') =E= 0;

PAIR3(I,J)..
    X(I,J,'2') - X(I,J,'3') =E= 0;

PAIR4(I,J)$(CTRLJ(J) NE 0)..
    X(I,J,'1') - Y(I,J+1) =E= 0;

REQ(J,K)..
    T(J,K) =G= D(K);

SELECT1(I)$(CTRLI(I) EQ 1)..
    Y(I,'1') =E= 1;
```

```
90
91 SELECT0(I)$(CTRLI(I) EQ 0)..
92     Y(I,'1') =E= 0;
93
94 MODEL SETCOVER /ALL/;
95 SOLVE SETCOVER USING LP MINIMIZING Z;
96 DISPLAY X.L,Y.L,T.L,TE.L;
97
98
```

```
GAMS 2.05 VAX VMS                    18-OCT-1990 11:18  PAGE    3
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
SYMBOL LISTING


SYMBOL     TYPE   REFERENCES

A          PARAM DECLARED      7 DEFINED      7     REF     71
CTRLI      PARAM DECLARED     23 DEFINED     23     REF     88
                 91
CTRLJ      PARAM DECLARED     24 DEFINED     24     REF     64
                 67     82
D          PARAM DECLARED     25 DEFINED     25     REF     86
FNODE      EQU   DECLARED     46 DEFINED     65 IMPL-ASN    95
                 REF     94
FNODEJ     EQU   DECLARED     47 DEFINED     68 IMPL-ASN    95
                 REF     94
GAIN       PARAM DECLARED     30 DEFINED     30     REF     65
                 68
I          SET   DECLARED      2 DEFINED      2     REF      7
                 23     26     33     36     46     47
                 49     50     51     52     54     55
               2*59     62   3*65   2*68   2*71   2*74
               2*77   2*80   2*83     88     89     91
                 92 CONTROL     40     41     59     62
                 64     67     71     73     76     79
                 82     88     91
J          SET   DECLARED      3 DEFINED      3     REF      7
                 24     33     34     35     36     46
                 47     48     49     51     52     53
                 59     64   3*65     67   2*68   4*71
               2*74   2*80     82   2*83     86
             CONTROL     40     41     59     64     67
                 70     73     79     82     85
K          SET   DECLARED      4 DEFINED      4     REF      7
                 25     27     33     34     35     48
                 49     53   2*59     65     68   4*71
                 74   2*86 CONTROL     40     59     65
                 68     70     73     85
M          PARAM DECLARED     26 DEFINED     26     REF     59
SETCOVER   MODEL DECLARED     94 DEFINED     94     REF     95
```

```
OBJ       EQU   DECLARED      44  DEFINED      59 IMPL-ASN      95
                REF      94
PAIR1     EQU   DECLARED      49  DEFINED      74 IMPL-ASN      95
                REF      94
PAIR2     EQU   DECLARED      50  DEFINED      77 IMPL-ASN      95
                REF      94
PAIR3     EQU   DECLARED      51  DEFINED      80 IMPL-ASN      95
                REF      94
PAIR4     EQU   DECLARED      52  DEFINED      83 IMPL-ASN      95
                REF      94
REQ       EQU   DECLARED      53  DEFINED      86 IMPL-ASN      95
                REF      94
SELECT0   EQU   DECLARED      55  DEFINED      92 IMPL-ASN      95
                REF      94
SELECT1   EQU   DECLARED      54  DEFINED      89 IMPL-ASN      95
                REF      94
SNODE     EQU   DECLARED      48  DEFINED      71 IMPL-ASN      95
                REF      94
```

```
GAMS 2.05 VAX VMS                    18-OCT-1990 11:18  PAGE     4
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
SYMBOL LISTING


SYMBOL    TYPE   REFERENCES

SRCE      EQU   DECLARED      45  DEFINED      62 IMPL-ASN      95
                REF      94
T         VAR   DECLARED      34 IMPL-ASN      95     REF      39
                71      86      96
TE        VAR   DECLARED      35 IMPL-ASN      95     REF      39
                59      71      96
W         PARAM DECLARED      27  DEFINED      27     REF      59
X         VAR   DECLARED      33 IMPL-ASN      95 ASSIGNED      40
                REF      39      65      68      71    2*74
                77    2*80      83      96
Y         VAR   DECLARED      36 IMPL-ASN      95 ASSIGNED      41
                REF      39      59      62    2*65      68
                77      83      89      92      96
Z         VAR   DECLARED      37 IMPL-ASN      95     REF      59
                95


SETS

I         LOCATIONS
J         STATES
K         SATELLITES


PARAMETERS
```

```
A        NUMBER OF OBSERVATION BLOCKS
CTRLI    SELECTION CONTROL
CTRLJ    EQUATION CONTROL
D        REQUIREMENTS VECTOR
GAIN     ARC GAIN
M        COST VECTOR
W        COST VECTOR
```

VARIABLES

```
T        SINK T IN-FLOW
TE       SINK TE IN-FLOW
X        BIPARTITE FLOW
Y        SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW
Z        OPTIMIZATION VARIABLE
```

EQUATIONS

```
FNODE     FACILITY NODE FLOW
FNODEJ    STATE J FACILITY NODE FLOW
OBJ       OBJECTIVE FUNCTION
PAIR1     EQUI DISTRIBUTION OF FLOW
PAIR2     EQUI DISTRIBUTION OF FLOW
PAIR3     EQUI DISTRIBUTION OF FLOW
```


SYMBOL LISTING

EQUATIONS

```
PAIR4     EQUI DISTRIBUTION OF FLOW
REQ       REQUIREMENTS
SELECT0   DO NOT SELECT FACILITY
SELECT1   SELECT FACILITY
SNODE     SATELLITE NODE FLOW
SRCE      SOURCE FLOW
```

MODELS

SETCOVER

COMPILATION TIME = 0.650 SECONDS

```
---- OBJ     =E=  OBJECTIVE FUNCTION

OBJ..  TE(1,1) + TE(1,2) + TE(1,3) + TE(2,1) + TE(2,2) + TE(2,3) + TE(3,1)

     + TE(3,2) + TE(3,3) - 1000*Y(1,1) - 1000*Y(2,1) - 1000*Y(3,1) + Z =E= 0 ;

---- SRCE     =G=  SOURCE FLOW

SRCE..  Y(1,1) + Y(2,1) + Y(3,1) =G= 1 ;

---- FNODE     =E=  FACILITY NODE FLOW

FNODE(1,1)..  X(1,1,1) + X(1,1,2) + X(1,1,3) - 4*Y(1,1) + Y(1,2) =E= 0 ;

FNODE(1,2)..  X(1,2,1) + X(1,2,2) + X(1,2,3) - 4*Y(1,2) + Y(1,3) =E= 0 ;

FNODE(2,1)..  X(2,1,1) + X(2,1,2) + X(2,1,3) - 4*Y(2,1) + Y(2,2) =E= 0 ;
REMAINING 3 ENTRIES SKIPPED

---- FNODEJ     =E=  STATE J FACILITY NODE FLOW

FNODEJ(1,3)..  X(1,3,1) + X(1,3,2) + X(1,3,3) - 3*Y(1,3) =E= 0 ;

FNODEJ(2,3)..  X(2,3,1) + X(2,3,2) + X(2,3,3) - 3*Y(2,3) =E= 0 ;

FNODEJ(3,3)..  X(3,3,1) + X(3,3,2) + X(3,3,3) - 3*Y(3,3) =E= 0 ;

---- SNODE     =E=  SATELLITE NODE FLOW

SNODE(1,1)..  - 8*X(1,1,1) - 3*X(2,1,1) - 2*X(3,1,1) + T(1,1) + TE(1,1) =E= 0 ;

SNODE(1,2)..  - 5*X(1,1,2) - 6*X(2,1,2) - 7*X(3,1,2) + T(1,2) + TE(1,2) =E= 0 ;
```

SNODE(1,3).. - 4*X(1,1,3) - 9*X(2,1,3) - 5*X(3,1,3) + T(1,3) + TE(1,3) =E= 0 ;

REMAINING 6 ENTRIES SKIPPED

EQUATION LISTING SOLVE SETCOVER USING LP FROM LINE 95

---- PAIR1 =E= EQUI DISTRIBUTION OF FLOW

PAIR1(1,1,2).. X(1,1,1) - X(1,1,2) =E= 0 ;

PAIR1(1,1,3).. X(1,1,1) - X(1,1,3) =E= 0 ;

PAIR1(1,2,2).. X(1,2,1) - X(1,2,2) =E= 0 ;

REMAINING 15 ENTRIES SKIPPED

---- PAIR2 =E= EQUI DISTRIBUTION OF FLOW

PAIR2(1).. X(1,1,1) - Y(1,1) =E= 0 ;

PAIR2(2).. X(2,1,1) - Y(2,1) =E= 0 ;

PAIR2(3).. X(3,1,1) - Y(3,1) =E= 0 ;

---- PAIR3 =E= EQUI DISTRIBUTION OF FLOW

PAIR3(1,1).. X(1,1,2) - X(1,1,3) =E= 0 ;

PAIR3(1,2).. X(1,2,2) - X(1,2,3) =E= 0 ;

PAIR3(1,3).. X(1,3,2) - X(1,3,3) =E= 0 ;

REMAINING 6 ENTRIES SKIPPED

```
---- PAIR4      =E=  EQUI DISTRIBUTION OF FLOW

PAIR4(1,1)..  X(1,1,1) - Y(1,2) =E= 0 ;

PAIR4(1,2)..  X(1,2,1) - Y(1,3) =E= 0 ;

PAIR4(2,1)..  X(2,1,1) - Y(2,2) =E= 0 ;

REMAINING 3 ENTRIES SKIPPED
```



```
---- REQ        =G=  REQUIREMENTS

REQ(1,1)..  T(1,1) =G= 6 ;

REQ(1,2)..  T(1,2) =G= 6 ;

REQ(1,3)..  T(1,3) =G= 6 ;

REMAINING 6 ENTRIES SKIPPED

---- SELECT1     =E=  SELECT FACILITY

          NONE

---- SELECT0     =E=  DO NOT SELECT FACILITY

          NONE
```

```
---- X         BIPARTITE FLOW


X(1,1,1)
              (.LO, .L, .UP = 0, 0, 1)
        1      FNODE(1,1)
       -8      SNODE(1,1)
        1      PAIR1(1,1,2)
        1      PAIR1(1,1,3)
        1      PAIR2(1)
        1      PAIR4(1,1)

X(1,1,2)
              (.LO, .L, .UP = 0, 0, 1)
        1      FNODE(1,1)
       -5      SNODE(1,2)
       -1      PAIR1(1,1,2)
        1      PAIR3(1,1)

X(1,1,3)
              (.LO, .L, .UP = 0, 0, 1)
        1      FNODE(1,1)
       -4      SNODE(1,3)
       -1      PAIR1(1,1,3)
       -1      PAIR3(1,1)

REMAINING 24 ENTRIES SKIPPED


---- T         SINK T IN-FLOW


T(1,1)
              (.LO, .L, .UP = 0, 0,  +INF)
        1      SNODE(1,1)
        1      REQ(1,1)

T(1,2)
              (.LO, .L, .UP = 0, 0,  +INF)
        1      SNODE(1,2)
        1      REQ(1,2)

T(1,3)
              (.LO, .L, .UP = 0, 0,  +INF)
        1      SNODE(1,3)
        1      REQ(1,3)

REMAINING 6 ENTRIES SKIPPED
```

```
---- TE        SINK TE IN-FLOW


TE(1,1)
            (.LO, .L, .UP = 0, 0,  +INF)
      1      OBJ
      1      SNODE(1,1)

TE(1,2)
            (.LO, .L, .UP = 0, 0,  +INF)
      1      OBJ
      1      SNODE(1,2)

TE(1,3)
            (.LO, .L, .UP = 0, 0,  +INF)
      1      OBJ
      1      SNODE(1,3)

REMAINING 6 ENTRIES SKIPPED


---- Y         SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW


Y(1,1)
            (.LO, .L, .UP = 0, 0, 1)
  -1000       OBJ
      1      SRCE
     -4      FNODE(1,1)
     -1      PAIR2(1)

Y(1,2)
            (.LO, .L, .UP = 0, 0, 1)
      1      FNODE(1,1)
     -4      FNODE(1,2)
     -1      PAIR4(1,1)

Y(1,3)
            (.LO, .L, .UP = 0, 0, 1)
      1      FNODE(1,2)
     -3      FNODEJ(1,3)
     -1      PAIR4(1,2)

REMAINING 6 ENTRIES SKIPPED
```

```
---- Z        OPTIMIZATION VARIABLE


Z
          (.LO, .L, .UP = -INF, 0, +INF)
      1    OBJ


GAMS 2.05 VAX VMS                    18-OCT-1990 11:18  PAGE    11
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
MODEL STATISTICS    SOLVE SETCOVER USING LP FROM LINE 95



MODEL STATISTICS

BLOCKS OF EQUATIONS    12      SINGLE EQUATIONS   65
BLOCKS OF VARIABLES     5      SINGLE VARIABLES   55
NON ZERO ELEMENTS     184

GENERATION TIME     =      0.890 SECONDS


EXECUTION TIME      =      1.440 SECONDS


GAMS 2.05 VAX VMS                    18-OCT-1990 11:18  PAGE    12
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
SOLUTION REPORT     SOLVE SETCOVER USING LP FROM LINE 95


          S O L V E    S U M M A R Y

   MODEL   SETCOVER          OBJECTIVE  Z
   TYPE    LP            DIRECTION  MINIMIZE
   SOLVER  BDMLP             FROM LINE  95

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE          1458.4444

 RESOURCE USAGE, LIMIT         0.520     1000.000
 ITERATION COUNT, LIMIT       24         1000

 BDM - LP  VERSION 1.01

 A. BROOKE, A. DRUD, AND A. MEERAUS,
 ANALYTIC SUPPORT UNIT,
 DEVELOPMENT RESEARCH DEPARTMENT,
 WORLD BANK,
```

WASHINGTON, D.C. 20433, U.S.A.

WORK SPACE NEEDED (ESTIMATE) -- 7567 WORDS.
WORK SPACE AVAILABLE -- 7567 WORDS.

EXIT -- OPTIMAL SOLUTION FOUND.

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU OBJ	.	.	.	1.000
---- EQU SRCE	1.000	1.481	+INF	.

OBJ OBJECTIVE FUNCTION
SRCE SOURCE FLOW

---- EQU FNODE FACILITY NODE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	-250.000
1.2	.	.	.	-139.417
2.1	.	.	.	-822.500
2.2	.	.	.	-205.625
3.1	.	.	.	-250.000
3.2	.	.	.	-219.917

---- EQU FNODEJ STATE J FACILITY NODE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.3	.	.	.	-113.568
2.3	.	.	.	-167.185
3.3	.	.	.	-86.259

---- EQU SNODE SATELLITE NODE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	-54.167
1.2	.	.	.	-1.000
1.3	.	.	.	-1.000
2.1	.	.	.	-1.000
2.2	.	.	.	-101.981
2.3	.	.	.	-1.000

3.1	.	.	.	-80.926
3.2	.	.	.	-1.000
3.3	.	.	.	-1.000

---- EQU PAIR1 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1.2	.	.	.	.
1.1.3	.	.	.	-491.000
1.2.2	.	.	.	64.546
1.2.3	.	.	.	-132.417
1.3.2	.	.	.	-105.568
1.3.3	.	.	.	-104.568
2.1.2	.	.	.	.
2.1.3	.	.	.	-1630.000
2.2.2	.	.	.	-99.306
2.2.3	.	.	.	.
2.3.2	.	.	.	-318.370
2.3.3	.	.	.	.
3.1.2	.	.	.	.
3.1.3	.	.	.	-488.000
3.2.2	.	.	.	177.056
3.2.3	.	.	.	.
3.3.2	.	.	.	.
3.3.3	.	.	.	-156.519

---- EQU PAIR2 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1	.	.	.	.
2	.	.	.	2290.000
3	.	.	.	.

---- EQU PAIR3 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	245.000
1.2	.	.	.	.

EQU PAIR3 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.3	.	.	.	.
2.1	.	.	.	816.500
2.2	.	.	.	-199.625
2.3	.	.	.	-158.185
3.1	.	.	.	243.000
3.2	.	.	.	-214.917
3.3	.	.	.	77.259

---- EQU PAIR4 EQUI DISTRIBUTION OF FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	.	.	307.667
1.2	.	.	.	201.287
2.1	.	.	.	.
2.2	.	.	.	295.931
3.1	.	.	.	629.667
3.2	.	.	.	38.861

---- EQU REQ REQUIREMENTS

	LOWER	LEVEL	UPPER	MARGINAL
1.1	6.000	6.000	+INF	54.167
1.2	6.000	6.000	+INF	1.000
1.3	6.000	6.000	+INF	1.000
2.1	6.000	6.000	+INF	1.000
2.2	6.000	6.000	+INF	101.981
2.3	6.000	6.000	+INF	1.000
3.1	6.000	6.000	+INF	80.926
3.2	6.000	6.000	+INF	1.000
3.3	6.000	6.000	+INF	1.000

---- EQU SELECT1 SELECT FACILITY

NONE

---- EQU SELECT0 DO NOT SELECT FACILITY

NONE

---- VAR X BIPARTITE FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1.1	.	0.444	1.000	.
1.1.2	.	0.444	1.000	.
1.1.3	.	0.444	1.000	.
1.2.1	.	0.444	1.000	.
1.2.2	.	0.444	1.000	.
1.2.3	.	0.444	1.000	.
1.3.1	.	0.444	1.000	.
1.3.2	.	0.444	1.000	.
1.3.3	.	0.444	1.000	.
2.1.1	.	0.370	1.000	.
2.1.2	.	0.370	1.000	.
2.1.3	.	0.370	1.000	.
2.2.1	.	0.370	1.000	.
2.2.2	.	0.370	1.000	.
2.2.3	.	0.370	1.000	.
2.3.1	.	0.370	1.000	.
2.3.2	.	0.370	1.000	.
2.3.3	.	0.370	1.000	.
3.1.1	.	0.667	1.000	.
3.1.2	.	0.667	1.000	.
3.1.3	.	0.667	1.000	.
3.2.1	.	0.667	1.000	.
3.2.2	.	0.667	1.000	.
3.2.3	.	0.667	1.000	.
3.3.1	.	0.667	1.000	.
3.3.2	.	0.667	1.000	.
3.3.3	.	0.667	1.000	.

---- VAR T SINK T IN-FLOW

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	6.000	+INF	.
1.2	.	6.000	+INF	.
1.3	.	6.000	+INF	.
2.1	.	6.000	+INF	.
2.2	.	6.000	+INF	.
2.3	.	6.000	+INF	.
3.1	.	6.000	+INF	.
3.2	.	6.000	+INF	.
3.3	.	6.000	+INF	.

```
---- VAR TE          SINK TE IN-FLOW

       LOWER    LEVEL    UPPER   MARGINAL

1.1     .        .       +INF    53.167
1.2     .      3.111     +INF      .
```



```
    VAR TE          SINK TE IN-FLOW

       LOWER    LEVEL    UPPER   MARGINAL

1.3     .      2.444     +INF      .
2.1     .      2.667     +INF      .
2.2     .        .       +INF    100.981
2.3     .      2.667     +INF      .
3.1     .        .       +INF    79.926
3.2     .      6.148     +INF      .
3.3     .      6.000     +INF      .

---- VAR Y          SOURCE OUT-FLOW AND INTERSTATE CONNECTOR FLOW

       LOWER    LEVEL    UPPER   MARGINAL

1.1     .      0.444     1.000     .
1.2     .      0.444     1.000     .
1.3     .      0.444     1.000     .
2.1     .      0.370     1.000     .
2.2     .      0.370     1.000     .
2.3     .      0.370     1.000     .
3.1     .      0.667     1.000     .
3.2     .      0.667     1.000     .
3.3     .      0.667     1.000     .

                    LOWER     LEVEL     UPPER    MARGINAL

---- VAR Z          -INF    1458.444    +INF       .

  Z         OPTIMIZATION VARIABLE


**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED
```

---- 96 VARIABLE X.L BIPARTITE FLOW

	1	2	3
1.1	0.444	0.444	0.444
1.2	0.444	0.444	0.444
1.3	0.444	0.444	0.444
2.1	0.370	0.370	0.370
2.2	0.370	0.370	0.370
2.3	0.370	0.370	0.370
3.1	0.667	0.667	0.667
3.2	0.667	0.667	0.667
3.3	0.667	0.667	0.667

---- 96 VARIABLE Y.L SOURCE OUT-FLOW AND INTERSTATE CONNECTOR
FLOW

	1	2	3
1	0.444	0.444	0.444
2	0.370	0.370	0.370
3	0.667	0.667	0.667

---- 96 VARIABLE T.L SINK T IN-FLOW

	1	2	3
1	6.000	6.000	6.000
2	6.000	6.000	6.000
3	6.000	6.000	6.000

---- 96 VARIABLE TE.L SINK TE IN-FLOW

	1	2	3
1		3.111	2.444
2	2.667		2.667
3		6.148	6.000

**** FILE SUMMARY
INPUT GSO91M:[PFORQUES]SETCOVR6.GMS;3
OUTPUT GSO91M:[PFORQUES]SETCOVR6.LIS;2
EXECUTION TIME = 1.080 SECONDS

```
MIP83 MAXSUM OUTPUT MAXSUM.OUT

 Copyright (C) 1986 by Sunset Software.
 All Rights Reserved Worldwide.
 1613 Chelsea Road, Suite 153
 San Marino, California 91108  U.S.A.

 Licensed Solely To: Wright-Patterson Air Force Base

..TITLE
 WEIGHTED-SUM MAX COVERAGE FORMULATION

..OBJECTIVE MAXIMIZE
 0.0 F1 - 1.0 F2

 *arc connectors
 + 0 [[XS1]] + 0 [[XS2]] + 0 [[XS3]]
 *state 1
 + 0 X110 + 0 X111 + 0 X112
 + 0 X210 + 0 X211 + 0 X212
 + 0 X310 + 0 X311 + 0 X312
 *state 2
 + 0 X413 + 0 X414 + 0 X415
 + 0 X513 + 0 X514 + 0 X515
 + 0 X613 + 0 X614 + 0 X615
 *state 3
 + 0 X716 + 0 X717 + 0 X718
 + 0 X816 + 0 X817 + 0 X818
 + 0 X916 + 0 X917 + 0 X918
 *interstate links
 + 0 X14 + 0 X25 + 0 X36
 + 0 X47 + 0 X58 + 0 X69
 *sink connectors
 + 0  X10T  + 0  X11T  + 0  X12T
 + 0  X13T  + 0  X14T  + 0  X15T
 + 0  X16T  + 0  X17T  + 0  X18T
 *select 'p' facilities
 + 0 p


..CONSTRAINTS

 OBJ1:
 *state 1
 + 8 X110 + 5 X111 + 4 X112
 + 3 X210 + 6 X211 + 9 X212
 + 2 X310 + 7 X311 + 5 X312
 *state 2
 + 6 X413 + 2 X414 + 7 X415
 + 9 X513 + 3 X514 + 6 X515
 + 4 X613 + 6 X614 + 5 X615
```

```
*state 3
+ 4 X716 + 8 X717 + 9 X718
+ 6 X816 + 7 X817 + 9 X818
+ 3 X916 + 9 X917 + 7 X918

- F1 = 0

OBJ2:
 12.223 XS1 + 13.556 XS2 + 11.111 XS3 - F2 = 0

NODE  S: XS1 + XS2 + XS3 - p = 0

NODE  1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE  2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE  3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE  4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE  5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE  6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE  7: X716 + X717 + X718 - 3 X47 = 0

NODE  8: X816 + X817 + X818 - 3 X58 = 0

NODE  9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T - X110 - X210 - X310 = 0

NODE 11: X11T - X111 - X211 - X311 = 0

NODE 12: X12T - X112 - X212 - X312 = 0

NODE 13: X13T - X413 - X513 - X613 = 0

NODE 14: X14T - X414 - X514 - X614 = 0

NODE 15: X15T - X415 - X515 - X615 = 0

NODE 16: X16T - X716 - X816 - X916 = 0

NODE 17: X17T - X717 - X817 - X917 = 0

NODE 18: X18T - X718 - X818 - X918 = 0

*---------------------------------------------------------*
* for the next constraint, coefficient of 'p' is (j*k)    *
* where                                                   *
*       j = number of states (months)                     *
*       k = number of demand locations (missions)         *
```

```
*-----------------------------------------------------*

NODE  T: - X10T - X11T - X12T
         - X13T - X14T - X15T
         - X16T - X17T - X18T + 9 p = 0

* select 'p' facilities
p = 1


* sink-connector flows
X10T - p = 0
X11T - p = 0
X12T - p = 0
X13T - p = 0
X14T - p = 0
X15T - p = 0
X16T - p = 0
X17T - p = 0
X18T - p = 0

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1  = 0
X110 - X14  = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2  = 0
X210 - X25  = 0


X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3  = 0
X310 - X36  = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47  = 0
X14  - X47  = 0

X513 - X514 = 0
X513 - X515 = 0
X514 - X515 = 0
X513 - X58  = 0
X25  - X58  = 0
```

X613 - X614 = 0
X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MAXIMIZED.
MIP Strategy: 1
Variables: 48
Integer: 3
Constraints: 74
0 LE, 74 EQ, 0 GE.
Non-zero LP elements: 227
Disk Space: 0K bytes.
Page Space: 28K bytes.
Capacity: 15.8% used.
Estimated Time: 00:00:44

Iter 47
Solution Time: 00:00:03
May have A L T E R N A T E S O L U T I O N

INTEGER SOLUTION

File: MAXSUM 11/25/90 17:14:40 Page 1-1
SOLUTION (Maximized): -11.1110 WEIGHTED-SUM MAX COVERAGE FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
F1	48.0000	0.0000	F2	11.1110	-1.0000
XS1	0.0000	0.0000	XS2	0.0000	0.0000
XS3	1.0000	0.0000	X110	0.0000	0.0000

X111	0.0000	0.0000	X112	0.0000	0.0000
X210	0.0000	0.0000	X211	0.0000	0.0000
X212	0.0000	0.0000	X310	1.0000	0.0000
X311	1.0000	0.0000	X312	1.0000	0.0000
X413	0.0000	0.0000	X414	0.0000	0.0000
X415	0.0000	0.0000	X513	0.0000	0.0000
X514	0.0000	0.0000	X515	0.0000	0.0000

File: MAXSUM 11/25/90 17:14:40 Page 1-2
SOLUTION (Maximized): -11.1110 WEIGHTED-SUM MAX COVERAGE FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
X613	1.0000	0.0000	X614	1.0000	0.0000
X615	1.0000	0.0000	X716	0.0000	0.0000
X717	0.0000	0.0000	X718	0.0000	0.0000
X816	0.0000	0.0000	X817	0.0000	0.0000
X818	0.0000	0.0000	X916	1.0000	0.0000
X917	1.0000	0.0000	X918	1.0000	0.0000
X14	0.0000	0.0000	X25	0.0000	0.0000
X36	1.0000	0.0000	X47	0.0000	0.0000
X58	0.0000	0.0000	X69	1.0000	0.0000
X10T	1.0000	0.0000	X11T	1.0000	0.0000

File: MAXSUM 11/25/90 17:14:40 Page 1-3
SOLUTION (Maximized): -11.1110 WEIGHTED-SUM MAX COVERAGE FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
X12T	1.0000	0.0000	X13T	1.0000	0.0000
X14T	1.0000	0.0000	X15T	1.0000	0.0000
X16T	1.0000	0.0000	X17T	1.0000	0.0000
X18T	1.0000	0.0000	p	1.0000	0.0000

File: MAXSUM 11/25/90 17:14:40 Page 1-4
CONSTRAINTS: WEIGHTED-SUM MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
OBJ1	0.0000 =	0.0000	OBJ2	0.0000 =	0.0000
NODE S	0.0000 =	0.0000	NODE 1	0.0000 =	0.0000

NODE 2	0.0000 =	0.0000	NODE 3	0.0000 =	0.0000
NODE 4	0.0000 =	0.0000	NODE 5	0.0000 =	0.0000
NODE 6	0.0000 =	0.0000	NODE 7	0.0000 =	0.0000
NODE 8	0.0000 =	0.0000	NODE 9	0.0000 =	0.0000
NODE 10	0.0000 =	0.0000	NODE 11	0.0000 =	0.0000
NODE 12	0.0000 =	0.0000	NODE 13	0.0000 =	0.0000
NODE 14	0.0000 =	0.0000	NODE 15	0.0000 =	0.0000
NODE 16	0.0000 =	0.0000	NODE 17	0.0000 =	0.0000

File: MAXSUM 11/25/90 17:14:40 Page 1-5
CONSTRAINTS: WEIGHTED-SUM MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE 18	0.0000 =	0.0000	NODE T	0.0000 =	0.0000
Row 23	1.0000 =	1.0000	Row 24	0.0000 =	0.0000
Row 25	0.0000 =	0.0000	Row 26	0.0000 =	0.0000
Row 27	0.0000 =	0.0000	Row 28	0.0000 =	0.0000
Row 29	0.0000 =	0.0000	Row 30	0.0000 =	0.0000
Row 31	0.0000 =	0.0000	Row 32	0.0000 =	0.0000
Row 33	0.0000 =	0.0000	Row 34	0.0000 =	0.0000
Row 35	0.0000 =	0.0000	Row 36	0.0000 =	0.0000
Row 37	0.0000 =	0.0000	Row 38	0.0000 =	0.0000
Row 39	0.0000 =	0.0000	Row 40	0.0000 =	0.0000

File: MAXSUM 11/25/90 17:14:40 Page 1-6
CONSTRAINTS: WEIGHTED-SUM MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 41	0.0000 =	0.0000	Row 42	0.0000 =	0.0000
Row 43	0.0000 =	0.0000	Row 44	0.0000 =	0.0000
Row 45	0.0000 =	0.0000	Row 46	0.0000 =	0.0000
Row 47	0.0000 =	0.0000	Row 48	0.0000 =	0.0000
Row 49	0.0000 =	0.0000	Row 50	0.0000 =	0.0000
Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000
Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000

Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

CONSTRAINTS: WEIGHTED-SUM MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000
Row 67	0.0000 =	0.0000	Row 68	0.0000 =	0.0000
Row 69	0.0000 =	0.0000	Row 70	0.0000 =	0.0000
Row 71	0.0000 =	0.0000	Row 72	0.0000 =	0.0000
Row 73	0.0000 =	0.0000	Row 74	0.0000 =	0.0000

Total Error: 0.000000

MIP83 MAXCON OUTPUT MAXCON.OUT


```
..TITLE
 CONSTRAINT-METHOD MAX COVERAGE FORMULATION

..OBJECTIVE MAXIMIZE
   F1

 *arc connectors
 + 0 [[XS1]]  + 0 [[XS2]]  + 0 [[XS3]]
 *state 1
 + 0 X110 + 0 X111 + 0 X112
 + 0 X210 + 0 X211 + 0 X212
 + 0 X310 + 0 X311 + 0 X312
 *state 2
 + 0 X413 + 0 X414 + 0 X415
 + 0 X513 + 0 X514 + 0 X515
 + 0 X613 + 0 X614 + 0 X615
 *state 3
 + 0 X716 + 0 X717 + 0 X718
 + 0 X816 + 0 X817 + 0 X818
 + 0 X916 + 0 X917 + 0 X918
 *interstate links
 + 0 X14 + 0 X25 + 0 X36
 + 0 X47 + 0 X58 + 0 X69
 *sink connectors
 + 0  X10T  + 0  X11T  + 0  X12T
 + 0  X13T  + 0  X14T  + 0  X15T
 + 0  X16T  + 0  X17T  + 0  X18T
 *select 'p' facilities
 + 0 p
 *variance criterion function
 + 0 F2

..CONSTRAINTS

 OBJ1:
 *state 1
 + 8 X110 + 5 X111 + 4 X112
 + 3 X210 + 6 X211 + 9 X212
 + 2 X310 + 7 X311 + 5 X312
 *state 2
 + 6 X413 + 2 X414 + 7 X415
 + 9 X513 + 3 X514 + 6 X515
```

+ 4 X613 + 6 X614 + 5 X615
*state 3
+ 4 X716 + 8 X717 + 9 X718
+ 6 X816 + 7 X817 + 9 X818
+ 3 X916 + 9 X917 + 7 X918

- F1 = 0

OBJ2:
12.223 XS1 + 13.556 XS2 + 11.111 XS3 - F2 = 0

SATISFICE: F2 <= 13.5560

NODE S: XS1 + XS2 + XS3 - p = 0

NODE 1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE 2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE 3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE 4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE 5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE 6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE 7: X716 + X717 + X718 - 3 X47 = 0

NODE 8: X816 + X817 + X818 - 3 X58 = 0

NODE 9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T - X110 - X210 - X310 = 0

NODE 11: X11T - X111 - X211 - X311 = 0

NODE 12: X12T - X112 - X212 - X312 = 0

NODE 13: X13T - X413 - X513 - X613 = 0

NODE 14: X14T - X414 - X514 - X614 = 0

NODE 15: X15T - X415 - X515 - X615 = 0

NODE 16: X16T - X716 - X816 - X916 = 0

NODE 17: X17T - X717 - X817 - X917 = 0

NODE 18: X18T - X718 - X818 - X918 = 0

```
*-------------------------------------------------------*
* for the next constraint, coefficient of 'p' is (j*k)
* where
*       j = number of states (months)
*       k = number of demand locations (missions)
*-------------------------------------------------------*

NODE  T: - X10T - X11T - X12T
       - X13T - X14T - X15T
       - X16T - X17T - X18T + 9 p = 0

* select 'p' facilities
p = 1


* sink-connector flows
X10T - p = 0
X11T - p = 0
X12T - p = 0
X13T - p = 0
X14T - p = 0
X15T - p = 0
X16T - p = 0
X17T - p = 0
X18T - p = 0

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1  = 0
X110 - X14  = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2  = 0
X210 - X25  = 0


X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3  = 0
X310 - X36  = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47  = 0
X14  - X47  = 0
```

X513 - X514 = 0
X513 - X515 = 0
X514 - X515 = 0
X513 - X58 = 0
X25 - X58 = 0

X613 - X614 = 0
X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MAXIMIZED.
MIP Strategy: 1
Variables: 48
Integer: 3
Constraints: 75
1 LE, 74 EQ, 0 GE.
Non-zero LP elements: 228
Disk Space: 0K bytes.
Page Space: 29K bytes.
Capacity: 15.9% used.
Estimated Time: 00:00:45

Iter 47
Solution Time: 00:00:04
May have A L T E R N A T E S O L U T I O N

INTEGER SOLUTION

File: MAXCON 11/25/90 17:15:06 Page 1-1
SOLUTION (Maximized): 58.0000 CONSTRAINT-METHOD MAX COVERAGE FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
F1	58.0000	1.0000	XS1	0.0000	0.0000
XS2	1.0000	0.0000	XS3	0.0000	0.0000
X110	0.0000	0.0000	X111	0.0000	0.0000
X112	0.0000	0.0000	X210	1.0000	0.0000
X211	1.0000	0.0000	X212	1.0000	0.0000
X310	0.0000	0.0000	X311	0.0000	0.0000
X312	0.0000	0.0000	X413	0.0000	0.0000
X414	0.0000	0.0000	X415	0.0000	0.0000
X513	1.0000	0.0000	X514	1.0000	0.0000
X515	1.0000	0.0000	X613	0.0000	0.0000

File: MAXCON 11/25/90 17:15:06 Page 1-2
SOLUTION (Maximized): 58.0000 CONSTRAINT-METHOD MAX COVERAGE FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
X614	0.0000	0.0000	X615	0.0000	0.0000
X716	0.0000	0.0000	X717	0.0000	0.0000
X718	0.0000	0.0000	X816	1.0000	0.0000
X817	1.0000	0.0000	X818	1.0000	0.0000
X916	0.0000	0.0000	X917	0.0000	0.0000
X918	0.0000	0.0000	X14	0.0000	0.0000
X25	1.0000	0.0000	X36	0.0000	0.0000
X47	0.0000	0.0000	X58	1.0000	0.0000
X69	0.0000	0.0000	X10T	1.0000	0.0000
X11T	1.0000	0.0000	X12T	1.0000	0.0000

File: MAXCON 11/25/90 17:15:06 Page 1-3
SOLUTION (Maximized): 58.0000 CONSTRAINT-METHOD MAX COVERAGE FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
X13T	1.0000	0.0000	X14T	1.0000	0.0000
X15T	1.0000	0.0000	X16T	1.0000	0.0000
X17T	1.0000	0.0000	X18T	1.0000	0.0000

p	1.0000	0.0000	F2	13.5560	0.0000

File: MAXCON 11/25/90 17:15:06 Page 1-4
CONSTRAINTS: CONSTRAINT-METHOD MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
OBJ1	0.0000 =	0.0000	OBJ2	0.0000 =	0.0000
SATISFIC	13.5560 <	13.5560	NODE S	0.0000 =	0.0000
NODE 1	0.0000 =	0.0000	NODE 2	0.0000 =	0.0000
NODE 3	0.0000 =	0.0000	NODE 4	0.0000 =	0.0000
NODE 5	0.0000 =	0.0000	NODE 6	0.0000 =	0.0000
NODE 7	0.0000 =	0.0000	NODE 8	0.0000 =	0.0000
NODE 9	0.0000 =	0.0000	NODE 10	0.0000 =	0.0000
NODE 11	0.0000 =	0.0000	NODE 12	0.0000 =	0.0000
NODE 13	0.0000 =	0.0000	NODE 14	0.0000 =	0.0000
NODE 15	0.0000 =	0.0000	NODE 16	0.0000 =	0.0000

File: MAXCON 11/25/90 17:15:06 Page 1-5
CONSTRAINTS: CONSTRAINT-METHOD MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE 17	0.0000 =	0.0000	NODE 18	0.0000 =	0.0000
NODE T	0.0000 =	0.0000	Row 24	1.0000 =	1.0000
Row 25	0.0000 =	0.0000	Row 26	0.0000 =	0.0000
Row 27	0.0000 =	0.0000	Row 28	0.0000 =	0.0000
Row 29	0.0000 =	0.0000	Row 30	0.0000 =	0.0000
Row 31	0.0000 =	0.0000	Row 32	0.0000 =	0.0000
Row 33	0.0000 =	0.0000	Row 34	0.0000 =	0.0000
Row 35	0.0000 =	0.0000	Row 36	0.0000 =	0.0000
Row 37	0.0000 =	0.0000	Row 38	0.0000 =	0.0000
Row 39	0.0000 =	0.0000	Row 40	0.0000 =	0.0000

File: MAXCON 11/25/90 17:15:06 Page 1-6
CONSTRAINTS: CONSTRAINT-METHOD MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 41	0.0000 =	0.0000	Row 42	0.0000 =	0.0000
Row 43	0.0000 =	0.0000	Row 44	0.0000 =	0.0000

Row 45	0.0000 =	0.0000	Row 46	0.0000 =	0.0000
Row 47	0.0000 =	0.0000	Row 48	0.0000 =	0.0000
Row 49	0.0000 =	0.0000	Row 50	0.0000 =	0.0000
Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000
Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000
Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

File: MAXCON 11/25/90 17:15:06 Page 1-7

CONSTRAINTS: CONSTRAINT-METHOD MAX COVERAGE FORMULATION

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000
Row 67	0.0000 =	0.0000	Row 68	0.0000 =	0.0000
Row 69	0.0000 =	0.0000	Row 70	0.0000 =	0.0000
Row 71	0.0000 =	0.0000	Row 72	0.0000 =	0.0000
Row 73	0.0000 =	0.0000	Row 74	0.0000 =	0.0000
Row 75	0.0000 =	0.0000			

Total Error: 0.000000

MIP83 P.LP output p.out


```
..TITLE
 CONSTRAINT METHOD SET COVERING FORMULATION (d=2)

..OBJECTIVE MINIMIZE
 1 F1

 *source connectors
 + 0 [[XS1]] + 0 [[XS2]] + 0 [[XS3]]
 *state 1
 + 0 X110 + 0 X111 + 0 X112
 + 0 X210 + 0 X211 + 0 X212
 + 0 X310 + 0 X311 + 0 X312
 *state 2
 + 0 X413 + 0 X414 + 0 X415
 + 0 X513 + 0 X514 + 0 X515
 + 0 X613 + 0 X614 + 0 X615
 *state 3
 + 0 X716 + 0 X717 + 0 X718
 + 0 X816 + 0 X817 + 0 X818
 + 0 X916 + 0 X917 + 0 X918
 *interstate links
 + 0 X14 + 0 X25 + 0 X36
 + 0 X47 + 0 X58 + 0 X69
 *demand-sink connectors
 +0 X10T +0 X11T +0 X12T
 +0 X13T +0 X14T +0 X15T
 +0 X16T +0 X17T +0 X18T
 *excess-sink connectors
 - 0 X10TE - 0 X11TE - 0 X12TE
 - 0 X13TE - 0 X14TE - 0 X15TE
 - 0 X16TE - 0 X17TE - 0 X18TE
 *variance criterion function
 + 0 F2

..BOUNDS
 *demand for mission 1 (nodes 10, 13, 16)
 X10T >= 2
 X13T >= 2
 X16T >= 2

 *demand for mission 2 (nodes 11, 14, 17)
```

```
X11T >= 2
X14T >= 2
X17T >= 2

*demand for mission 3 (nodes 12, 15, 18)
X12T >= 2
X15T >= 2
X18T >= 2


..CONSTRAINTS

OBJ1:
 1000 XS1 + 1000 XS2 + 1000 XS3
 - 1  X10TE  - 1  X11TE  - 1  X12TE
 - 1  X13TE  - 1  X14TE  - 1  X15TE
 - 1  X16TE  - 1  X17TE  - 1  X18TE
 - F1 = 0

OBJ2:
 12.223 XS1 + 13.556 XS2 + 11.111 XS3 - F2 = 0

SATISFICE:
 F2 <= 12.0884

NODE  S: XS1 + XS2 + XS3 >= 1

NODE  1: X110 + X111 + X112 + X14 - 4 XS1 = 0

NODE  2: X210 + X211 + X212 + X25 - 4 XS2 = 0

NODE  3: X310 + X311 + X312 + X36 - 4 XS3 = 0

NODE  4: X413 + X414 + X415 + X47 - 4 X14 = 0

NODE  5: X513 + X514 + X515 + X58 - 4 X25 = 0

NODE  6: X613 + X614 + X615 + X69 - 4 X36 = 0

NODE  7: X716 + X717 + X718 - 3 X47 = 0

NODE  8: X816 + X817 + X818 - 3 X58 = 0

NODE  9: X916 + X917 + X918 - 3 X69 = 0

NODE 10: X10T + X10TE - 8 X110 - 3 X210 - 2 X310 = 0

NODE 11: X11T + X11TE - 5 X111 - 6 X211 - 7 X311 = 0

NODE 12: X12T + X12TE - 4 X112 - 9 X212 - 5 X312 = 0

NODE 13: X13T + X13TE - 6 X413 - 9 X513 - 4 X613 = 0
```

NODE 14: X14T + X14TE - 2 X414 - 3 X514 - 6 X614 = 0

NODE 15: X15T + X15TE - 7 X415 - 6 X515 - 5 X615 = 0

NODE 16: X16T + X16TE - 4 X716 - 6 X816 - 3 X916 = 0

NODE 17: X17T + X17TE - 8 X717 - 7 X817 - 9 X917 = 0

NODE 18: X18T + X18TE - 9 X718 - 9 X818 - 7 X918 = 0

NODE T: - X10T - X11T - X12T
- X13T - X14T - X15T
- X16T - X17T - X18T <= 0

NODE TE: - X10TE - X11TE - X12TE
- X13TE - X14TE - X15TE
- X16TE - X17TE - X18TE <= 0

* equi-distribution of flow at each location
X110 - X111 = 0
X110 - X112 = 0
X111 - X112 = 0
X110 - XS1 = 0
X110 - X14 = 0

X210 - X211 = 0
X210 - X212 = 0
X211 - X212 = 0

X210 - XS2 = 0
X210 - X25 = 0

X310 - X311 = 0
X310 - X312 = 0
X311 - X312 = 0
X310 - XS3 = 0
X310 - X36 = 0

X413 - X414 = 0
X413 - X415 = 0
X414 - X415 = 0
X413 - X47 = 0
X14 - X47 = 0

X513 - X514 = 0
X513 - X515 = 0
X514 - X515 = 0
X513 - X58 = 0
X25 - X58 = 0

X613 - X614 = 0
X613 - X615 = 0
X614 - X615 = 0
X613 - X69 = 0
X36 - X69 = 0

X716 - X717 = 0
X716 - X718 = 0
X717 - X718 = 0
X716 - X47 = 0

X816 - X817 = 0
X816 - X818 = 0
X817 - X818 = 0
X816 - X58 = 0

X916 - X917 = 0
X916 - X918 = 0
X917 - X918 = 0
X916 - X69 = 0

Statistics-
MIP83 Version 5.00a
Machine memory: 256K bytes.
Pagable memory: 0K bytes.
Objective Function is MINIMIZED.
MIP Strategy: 1
Variables: 56
Integer: 3
Constraints: 66
3 LE, 62 EQ, 1 GE.
Non-zero LP elements: 210
Disk Space: 0K bytes.
Page Space: 30K bytes.
Capacity: 15.5% used.
Estimated Time: 00:00:36

Iter 53
Solution Time: 00:00:03
May have A L T E R N A T E S O L U T I O N

Optimal Solution: 965.6052 Max Node Depth: 546 Limit: NONE

Solution: 970.0000 Iter: 3 Nodes: 2 Iteration Time: 00:00:01
INTEGER SOLUTION

File: P 11/23/90 17:32:32 Page 1-1
SOLUTION (Minimized): 970.0000 CONSTRAINT METHOD SET COVERING FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
F1	970.0000	1.0000	XS1	0.0000	0.0000
XS2	0.0000	0.0000	XS3	1.0000	0.0000
X110	0.0000	0.0000	X111	0.0000	0.0000
X112	0.0000	0.0000	X210	0.0000	0.0000
X211	0.0000	0.0000	X212	0.0000	0.0000
X310	1.0000	0.0000	X311	1.0000	0.0000
X312	1.0000	0.0000	X413	0.0000	0.0000
X414	0.0000	0.0000	X415	0.0000	0.0000
X513	0.0000	0.0000	X514	0.0000	0.0000
X515	0.0000	0.0000	X613	1.0000	0.0000

File: P 11/23/90 17:32:32 Page 1-2
SOLUTION (Minimized): 970.0000 CONSTRAINT METHOD SET COVERING FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
X614	1.0000	0.0000	X615	1.0000	0.0000
X716	0.0000	0.0000	X717	0.0000	0.0000
X718	0.0000	0.0000	X816	0.0000	0.0000
X817	0.0000	0.0000	X818	0.0000	0.0000
X916	1.0000	0.0000	X917	1.0000	0.0000
X918	1.0000	0.0000	X14	0.0000	0.0000
X25	0.0000	0.0000	X36	1.0000	0.0000
X47	0.0000	0.0000	X58	0.0000	0.0000
X69	1.0000	0.0000	X10T	2.0000	0.0000
X11T	2.0000	0.0000	X12T	2.0000	0.0000

File: P 11/23/90 17:32:32 Page 1-3
SOLUTION (Minimized): 970.0000 CONSTRAINT METHOD SET COVERING FORMULATION

Variable	Activity	Cost	Variable	Activity	Cost
X13T	2.0000	0.0000	X14T	2.0000	0.0000
X15T	2.0000	0.0000	X16T	2.0000	0.0000

X17T	2.0000	0.0000	X18T	2.0000	0.0000
X10TE	0.0000	0.0000	X11TE	5.0000	0.0000
X12TE	3.0000	0.0000	X13TE	2.0000	0.0000
X14TE	4.0000	0.0000	X15TE	3.0000	0.0000
X16TE	1.0000	0.0000	X17TE	7.0000	0.0000
X18TE	5.0000	0.0000	F2	11.1110	0.0000

File: P 11/23/90 17:32:32 Page 1-4

CONSTRAINTS: CONSTRAINT METHOD SET COVERING FORMULATION (d=2)

Constraint	Activity	RHS	Constraint	Activity	RHS
OBJ1	0.0000 =	0.0000	OBJ2	0.0000 =	0.0000
SATISFIC	11.1110 <	12.0884	NODE S	1.0000 >	1.0000
NODE 1	0.0000 =	0.0000	NODE 2	0.0000 =	0.0000
NODE 3	0.0000 =	0.0000	NODE 4	0.0000 =	0.0000
NODE 5	0.0000 =	0.0000	NODE 6	0.0000 =	0.0000
NODE 7	0.0000 =	0.0000	NODE 8	0.0000 =	0.0000
NODE 9	0.0000 =	0.0000	NODE 10	0.0000 =	0.0000
NODE 11	0.0000 =	0.0000	NODE 12	0.0000 =	0.0000
NODE 13	0.0000 =	0.0000	NODE 14	0.0000 =	0.0000
NODE 15	0.0000 =	0.0000	NODE 16	0.0000 =	0.0000

File: P 11/23/90 17:32:32 Page 1-5

CONSTRAINTS: CONSTRAINT METHOD SET COVERING FORMULATION (d=2)

Constraint	Activity	RHS	Constraint	Activity	RHS
NODE 17	0.0000 =	0.0000	NODE 18	0.0000 =	0.0000
NODE T	-18.0000 <	0.0000	NODE TE	-30.0000 <	0.0000
Row 25	0.0000 =	0.0000	Row 26	0.0000 =	0.0000
Row 27	0.0000 =	0.0000	Row 28	0.0000 =	0.0000
Row 29	0.0000 =	0.0000	Row 30	0.0000 =	0.0000
Row 31	0.0000 =	0.0000	Row 32	0.0000 =	0.0000
Row 33	0.0000 =	0.0000	Row 34	0.0000 =	0.0000
Row 35	0.0000 =	0.0000	Row 36	0.0000 =	0.0000

Row 37	0.0000 =	0.0000	Row 38	0.0000 =	0.0000
Row 39	0.0000 =	0.0000	Row 40	0.0000 =	0.0000

File: P 11/23/90 17:32:32 Page 1-6

CONSTRAINTS: CONSTRAINT METHOD SET COVERING FORMULATION (d=2)

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 41	0.0000 =	0.0000	Row 42	0.0000 =	0.0000
Row 43	0.0000 =	0.0000	Row 44	0.0000 =	0.0000
Row 45	0.0000 =	0.0000	Row 46	0.0000 =	0.0000
Row 47	0.0000 =	0.0000	Row 48	0.0000 =	0.0000
Row 49	0.0000 =	0.0000	Row 50	0.0000 =	0.0000
Row 51	0.0000 =	0.0000	Row 52	0.0000 =	0.0000
Row 53	0.0000 =	0.0000	Row 54	0.0000 =	0.0000
Row 55	0.0000 =	0.0000	Row 56	0.0000 =	0.0000
Row 57	0.0000 =	0.0000	Row 58	0.0000 =	0.0000
Row 59	0.0000 =	0.0000	Row 60	0.0000 =	0.0000

File: P 11/23/90 17:32:32 Page 1-7

CONSTRAINTS: CONSTRAINT METHOD SET COVERING FORMULATION (d=2)

Constraint	Activity	RHS	Constraint	Activity	RHS
Row 61	0.0000 =	0.0000	Row 62	0.0000 =	0.0000
Row 63	0.0000 =	0.0000	Row 64	0.0000 =	0.0000
Row 65	0.0000 =	0.0000	Row 66	0.0000 =	0.0000

Total Error: 0.000000

```
*************************************************************
*          GEODSS OPTIMAL LOCATION SOLVER                   *
*************************************************************
* Programmer: Maj Pierre Forgues
* Date      : January 1991
*
* Purpose  : Prioritize one-site, two-site, three-site
*          choice for locating GEODSS given a set of
*          12 candidate locations and 6 satellites
*
* Algorithm : This multicriteria decision making problem
*          involves two objectives: maximize the number
*          of observations and minimize the variance
*          in the number of observations collected on
*          each satellite. This ensures that each
*          orbit is covered equally well.
*
*          The value of each of the above objectives is
*          calculated for each feasible alternative for
*          each month of the year. The alternatives are
*          ranked based on the deviation from the ideal.
*
* Variables : - Reals -
*             PROB    : probability of observing
*             BLOCKS  : number of 5 mins blocks
*             EXPECT  : expected value of useable blocks
*             OBJ1    : value of objective 1 (max obs)
*             OBJ2    : value of objective 2 (min var)
*             ONEDEV  : deviation from ideal (p=1)
*             TWODEV  : deviation from ideal (p=2)
*             TREDEV  : deviation from ideal (p=3)
*             IDLP1   : y-space coord of ideal (p=1)
*             IDLP2   : y-space coord of ideal (p=2)
*             IDLP3   : y-space coord of ideal (p=3)
*           - Logicals -
*             FEASIB  : record of feasibility
*             NSET    : record of N-points
*           - Integers -
*             ONEPRM  : permutation vector (p=1)
*             TWOPRM  : permutation vector (p=2)
*             TREPRM  : permutation vector (p=3)
*             FEA1    : no. of feasible p=1 alt.
*             FEA2    : no. of feasible p=2 alt.
*             FEA3    : no. of feasible p=3 alt.
*             EFF1    : no. in N-set for p=1
*             EFF2    : no. in N-set for p=2
*             EFF3    : no. in N-set for p=3
*
* Subprograms:
*          PROCAL  - returns values of PROB(I,J,K)
*          EXPVAL  - returns values of EXPECT(I,J,K)
```

```
*         FEACHK  - identifies feasible alternatives
*         OBJCAL  - returns values of OBJ1 and OBJ2
*         IDLCAL  - returns values of IDLP1/2/3
*         EFFSET  - identifies set of N-points
*         DEVCAL  - computes deviations from ideal
*         PRIORI  - prioritizes alternatives
*         PRTOUT  - prints results
*         UTILS   - convert to utils
****************************************************************


C***************************************************************
C
C          MAIN PROGRAM BEGINS HERE
C
C***************************************************************


C
C   Variable Declarations
C
C
      REAL*8 PROB(12,12,6),BLOCKS(12),EXPECT(12,12,6)

      REAL*8 OBJ1(298),OBJ2(298)

      REAL*8 IDLP1(2),IDLP2(2),IDLP3(2)

      REAL*8 ONEDEV(12),TWODEV(66),TREDEV(220)

      INTEGER ONEPRM(12),TWOPRM(66),TREPRM(220)

      INTEGER FEA1,FEA2,FEA3,EFF1,EFF2,EFF3

      LOGICAL FEASIB(298),NSET(298)
C
C
C   Compute total number of 5 minute blocks for each month
C
C
      BLOCKS(1)  = (31.)*(24.)*(60.)/(5.)
      BLOCKS(2)  = (28.)*(24.)*(60.)/(5.)
      BLOCKS(3)  = (31.)*(24.)*(60.)/(5.)
      BLOCKS(4)  = (30.)*(24.)*(60.)/(5.)
      BLOCKS(5)  = (31.)*(24.)*(60.)/(5.)
      BLOCKS(6)  = (30.)*(24.)*(60.)/(5.)
      BLOCKS(7)  = (31.)*(24.)*(60.)/(5.)
      BLOCKS(8)  = (31.)*(24.)*(60.)/(5.)
      BLOCKS(9)  = (30.)*(24.)*(60.)/(5.)
      BLOCKS(10) = (31.)*(24.)*(60.)/(5.)
      BLOCKS(11) = (30.)*(24.)*(60.)/(5.)
      BLOCKS(12) = (31.)*(24.)*(60.)/(5.)
C
```

```
C
C   Compute probability of observable blocks at location
C   I, in month J, for satellite K
C
C

    CALL PROCAL(PROB)

C
C
C   Compute expected value of the number of useable
C   blocks at location I, in month J, for satellite K
C
C

    CALL EXPVAL(EXPECT,BLOCKS,PROB)

C
C
C   Determine feasibility of all possible alternatives
C   for the p=1,2, and 3 problems
C
C

    CALL FEACHK(EXPECT,FEASIB,FEA1,FEA2,FEA3)

C
C
C   For each feasible alternatives, compute the value
C   of each objective function
C
C

    CALL OBJCAL(EXPECT,FEASIB,OBJ1,OBJ2)

C
C
C   Determine which alternatives are non-dominated
C   (i.e. generate the set of efficient solutions)
C
C

    CALL EFFSET(FEASIB,OBJ1,OBJ2,NSET,EFF1,EFF2,EFF3)

C
C
C   Determine Y-space coordinates of ideal solution
C
C

    CALL IDLCAL(OBJ1,OBJ2,IDLP1,IDLP2,IDLP3)
```

```
C
C
C   Convert OBJ1 and OBJ2 to utils
C
C

      CALL UTILS(OBJ1,OBJ2,FEASIB)

C
C
C   Compute deviation from ideal for feasible points
C
C

      CALL DEVCAL(OBJ1,OBJ2,FEASIB,
     +           ONEDEV,TWODEV,TREDEV)

C
C
C   Rank alternatives for the p=1,2, and 3 problems
C   according to minimum deviation from ideal
C
C

      CALL PRIORI(ONEDEV,TWODEV,TREDEV,ONEPRM,TWOPRM,TREPRM)

C
C
C   Print out results in file 'RESULTS.OUT'
C
C

      CALL PRTOUT(EXPECT,OBJ1,OBJ2,ONEDEV,TWODEV,TREDEV,
     +           ONEPRM,TWOPRM,TREPRM,IDLP1,IDLP2,IDLP3,
     +           FEA1,FEA2,FEA3,EFF1,EFF2,EFF3)

      END

C**********************************************************
C
C            MAIN PROGRAM ENDS HERE
C
C**********************************************************

**********************************************************
*                SUBPROGRAM PROCAL
**********************************************************
*
* Purpose  : Read probability input data and compute
*            PROB(I,J,K), the probability a 5 mins block
*            is useful for observing satellite K, at
*            location I, in month J.
```

```
*
* Variables : - Reals -
*            PROB  : as described above
*            PROBA : probability event A will occur
*            PROBB :     "     event B "    "
*            PROBD :     "     event D "    "
*            PROBE :     "     event E "    "
*            PROBF :     "     event F "    "
*            PROBG :     "     event G "    "
*          - Integers -
*            I    : loop counter (location number)
*            J    :  "     "  (month number)
*
*            K    :  "     "  (satellite number)
*          - Characters -
*            HEADER: table header string used to
*                  separate tables in data files
*                  PROBE.DAT, PROBF.DAT, PROBG.DAT
*
* Events   : A. The sun is a least 6 deg below horizon
*          B. Surface wind speeds are less than 25 kts
*          C. Temperature is warmer than -50C
*          D. Satellite is at least 15 deg above horizon
*          E. Five minute CFLOS
*          F. Satellite is illuminated
*
**************************************************************

      SUBROUTINE PROCAL(PROB)

      REAL*8 PROB(12,12,6)
      REAL*8 PROBA(12,12),PROBB(12,12),PROBD(12,6)
      REAL*8 PROBE(12,12,6),PROBF(12,12,6),PROBC(12,12)
      INTEGER I,J,K
      CHARACTER*80 HEADER

      OPEN (UNIT=10,FILE='PROBA.DAT',STATUS='OLD')
      OPEN (UNIT=20,FILE='PROBB.DAT',STATUS='OLD')
      OPEN (UNIT=30,FILE='PROBD.DAT',STATUS='OLD')
      OPEN (UNIT=40,FILE='PROBE.DAT',STATUS='OLD')
      OPEN (UNIT=50,FILE='PROBF.DAT',STATUS='OLD')
      OPEN (UNIT=60,FILE='PROBC.DAT',STATUS='OLD')

C     Read PROBA.DAT, PROBB.DAT, and PROBC.DAT files

      DO 1000 I=1,12

          READ(10,1600) (PROBA(I,J),J=1,12)
          READ(20,1600) (PROBB(I,J),J=1,12)
          READ(60,1600) (PROBC(I,J),J=1,12)

 1000 CONTINUE
```

```
C    Read PROBD.DAT file

     DO 1100 I=1,12

         READ(30,1600) (PROBD(I,K),K=1,6)

1100 CONTINUE

C    Read PROBE.DAT file, beginning with first HEADER

     READ(40,'(A80)') HEADER

     DO 1300 K=1,6
       DO 1200 I=1,12

         READ(40,1600) (PROBE(I,J,K),J=1,12)

1200   CONTINUE

       READ(40,'(A80)') HEADER

1300 CONTINUE

C    Read PROBF.DAT beginning with header

     READ(50,'(A80)') HEADER

     DO 1500 K=1,6
       DO 1400 I=1,12

         READ(50,1600) (PROBF(I,J,K),J=1,12)

1400   CONTINUE

       READ(50,'(A80)') HEADER

1500 CONTINUE

1600 FORMAT (12F4.2)

C
C
C    Compute the probability, PROB(I,J,K), a given
C    5 min block is useable for observation purposes
C
C
     DO 2000 I=1,12
       DO 1900 J=1,12
         DO 1800 K=1,6

           PROB(I,J,K)=PROBA(I,J)*PROBB(I,J)*PROBD(I,K)*
    +               PROBE(I,J,K)*PROBF(I,J,K)*
    +               PROBC(I,J)
```

```
1800      CONTINUE
1900    CONTINUE
2000  CONTINUE


      CLOSE(10)
      CLOSE(20)
      CLOSE(30)
      CLOSE(40)
      CLOSE(50)
      CLOSE(60)

      RETURN

      END
************************************************************
*                SUBPROGRAM EXPVAL
************************************************************
*
* Purpose   : Compute the expected value of the number
*             of useable 5 mins blocks to observe satellite
*             K, at location I, in month J.  A block is
*             considered "useable" when all conditions
*             necessary for observation are met.
*
* Variables :  - Reals -
*                EXPECT: expected value
*                BLOCKS: total number of 5 mins blocks
*                PROB  : probability a block is useable

*              - Integers -
*                I    : loop counter (location number)
*                J    :  "     "   (month number)
*
*                K    :  "     "   (satellite number)
*
************************************************************

      SUBROUTINE EXPVAL(EXPECT,BLOCKS,PROB)

      REAL*8 EXPECT(12,12,6),BLOCKS(12),PROB(12,12,6)
      INTEGER I,J,K

      DO 1900 I=1,12
        DO 1800 J=1,12
          DO 1700 K=1,6

            EXPECT(I,J,K) = (PROB(I,J,K)) * (BLOCKS(J))

1700      CONTINUE
1800    CONTINUE
1900  CONTINUE
```

```
      RETURN

      END
**************************************************************
*                 SUBPROGRAM FEACHK
**************************************************************
*
* Purpose  : Determine which alternatives are feasible
*
* Variables : - Reals -
*              EXPECT: expected value (number of blocks)
*              OBSREQ: monthly observation requirement
*              OBS   : observations for alternative X
*                      in month J on satellite K
*            - Integers -
*              I     : loop counter (location number)
*              J     :  "      "   (month number)
*              K     :  "      "   (satellite number)
*              X     : alternative number
*              SITE1 :  "      "
*              SITE2 :  "      "
*              SITE3 :  "      "
*              FEA1  : no. of feasible p=1 alternatives
*              FEA2  : no. of feasible p=2 alternatives
*              FEA3  : no. of feasible p=3 alternatives
*            - Logicals -
*              FEASIB: feasibility record
*
**************************************************************

      SUBROUTINE FEACHK(EXPECT,FEASIB,FEA1,FEA2,FEA3)

      REAL*8 EXPECT(12,12,6),OBSREQ(6),OBS(298,13,6)

      INTEGER I,J,K,X,SITE1,SITE2,SITE3

      INTEGER FEA1,FEA2,FEA3

      LOGICAL FEASIB(298)
C
C
C     Read monthly observation requirement for each satel
C     lite
C
C
      OPEN (UNIT=70,FILE='OBSREQ.DAT',STATUS='OLD')
      READ (70,2000) (OBSREQ(K),K=1,6)
2000  FORMAT (6F5.0)
      CLOSE(70)

C
C
C     Initialize the FEASIB vector
```

```
C
C

      DO 2001 X=1,298
        FEASIB(X)=.TRUE.
2001  CONTINUE


C
C
C    Perform feasibility check for p=1 alternatives
C
C
      DO 2100 K=1,6
        DO 2090 I=1,12
          DO 2080 J=1,12

            OBS(I,J,K)=EXPECT(I,J,K)
            IF (OBS(I,J,K).LT.OBSREQ(K))
     +       FEASIB(I)=.FALSE.

2080        CONTINUE
2090      CONTINUE
2100  CONTINUE


C
C
C    Perform feasibility check for p=2 alternatives
C
C

      DO 2200 K=1,6

        X=12

        DO 2190 SITE1=1,11
          DO 2180 SITE2=(SITE1+1),12

            X=X+1

            DO 2170 J=1,12

              OBS(X,J,K)=EXPECT(SITE1,J,K)+
     +                  EXPECT(SITE2,J,K)

              IF (OBS(X,J,K).LT.OBSREQ(K))
     +         FEASIB(X)=.FALSE.


2170          CONTINUE
2180        CONTINUE
2190      CONTINUE
```

```
2200  CONTINUE

C
C
C   Perform feasibility check for p=3 alternatives
C
C

    DO 2300 K=1,6

      X=78

      DO 2290 SITE1=1,10
        DO 2280 SITE2=(SITE1+1),11
          DO 2270 SITE3=(SITE2+1),12

            X=X+1

            DO 2260 J=1,12

              OBS(X,J,K)=EXPECT(SITE1,J,K)+
     +                   EXPECT(SITE2,J,K)+
     +                   EXPECT(SITE3,J,K)

              IF (OBS(X,J,K).LT.OBSREQ(K))
     +         FEASIB(X)=.FALSE.

2260          CONTINUE
2270        CONTINUE
2280      CONTINUE
2290    CONTINUE
2300  CONTINUE

C
C
C   Print OBS(X,J,K) to file
C
C

    OPEN(UNIT=91,FILE='ALTOBS.OUT',STATUS='NEW')
    DO 2305 K=1,6

      WRITE(91,*)
      WRITE(91,'(1X,A,I2)') ('OBSERVATIONS ON SATNO= ',K)
      WRITE(91,'(1X,A)') 'ALTERNATIVES (ROWS) MONTHS
     +                (COLS)'
      WRITE(91,'(1X,A)') '(ALT NO IN LAST COLUMN)'
      WRITE(91,*)

      DO 2304 X=1,298
```

```
        OBS(X,13,K)=DBLE(X)
        WRITE(91,'(13(F6.0))') (OBS(X,J,K),J=1,13)

2304    CONTINUE
2305  CONTINUE

C
C
C    Print FEASIB(X) to file
C
C
      OPEN(UNIT=93,FILE='FEASIB.OUT',STATUS='NEW')
      WRITE(93,*)
      WRITE(93,'(1X,A)') 'FEASIBILITY LIST'

      DO 2306 J=1,295,5
        WRITE(93,'(5(3X,I5,L3))')((X,FEASIB(X)),X=J,J+4)
2306  CONTINUE
      WRITE(93,'(3(3X,I5,L3))')((X,FEASIB(X)),X=296,298)


      CLOSE(91)
      CLOSE(93)

C
C
C    Count number of feasible alternatives
C
C
      FEA1=0
      FEA2=0
      FEA3=0

      DO 2400 X=1,12
        IF (FEASIB(X)) FEA1=FEA1+1
2400  CONTINUE

      DO 2500 X=13,78
        IF (FEASIB(X)) FEA2=FEA2+1
2500  CONTINUE

      DO 2600 X=79,298
        IF (FEASIB(X)) FEA3=FEA3+1
2600  CONTINUE


      RETURN
      END
***********************************************************
*                 SUBPROGRAM OBJCAL
***********************************************************
*
* Purpose  : Calculate the value of both objective
```

```
*          functions for all feasible alternatives
*
* Variables : - Reals -
*            EXPECT: expected value (number of blocks)
*            OBJ1  : value of first objective function
*                  where OBJ1(X) is the total number
*                  of observations collected on all
*                  satellites by alternative X
*                  summed
*                  over all months
*            OBJ2  : value of second objective func
*                  tion
*                  where OBJ2(X) is the sum over all
*                  months of the monthly variance in
*                  the number of observations col
*                  lected
*                  on each satellite
*            MEAN  : mean number of observations
*            SUMDEV: sum of squared deviation from
*                  mean
*            MONTH : observations in current month
*
*          - Integers -
*            I    : loop counter (location number)
*            J    :  "     "  (month number)
*            K    :  "     "  (satellite number)
*            X    : alternative number
*            SITE1 :  "    "
*            SITE2 :  "    "
*            SITE3 :  "    "
*          - Logicals -
*            FEASIB: feasibility record
*
**********************************************************

      SUBROUTINE OBJCAL(EXPECT,FEASIB,OBJ1,OBJ2)


      REAL*8 OBJ1(298),OBJ2(298),EXPECT(12,12,6)

      REAL*8 MEAN,SUMDEV,MONTH

      INTEGER I,J,K,X,SITE1,SITE2,SITE3

      LOGICAL FEASIB(298)

C
C
C     Perform computations for p=1 alternatives
C
C
      X=0
      DO 3100 I=1,12
```

```
      X=X+1
      OBJ1(X)=0
      OBJ2(X)=0

      IF (.NOT.FEASIB(X)) THEN

         OBJ2(X)=1.0D20
         GO TO 3090

      END IF


      DO 3080 J=1,12

C          Set current month observation total to zero
        MONTH=0

        DO 3060 K=1,6

          MONTH=MONTH+EXPECT(I,J,K)


3059    FORMAT(3(1X,A,I3),2(1X,A,F10.0))

3060      CONTINUE

        OBJ1(X)=OBJ1(X)+MONTH
        MEAN=MONTH/6
        SUMDEV=0

        DO 3070 K=1,6

          SUMDEV=SUMDEV+((EXPECT(I,J,K)-MEAN)**2)


3069    FORMAT(1X,3(1X,A,I2),2(1X,A,F10.0))

3070      CONTINUE

        OBJ2(X)=OBJ2(X)+(SUMDEV/6)

3080    CONTINUE
3090    CONTINUE
3100  CONTINUE

C
C
C    Perform computations for p=2 alternatives
C
C
     X=12
     DO 3200 SITE1=1,11
```

```
      DO 3190 SITE2=(SITE1+1),12

        X=X+1
        OBJ1(X)=0
        OBJ2(X)=0

        IF (.NOT.FEASIB(X)) THEN

          OBJ2(X)=1.0D20
          GO TO 3180

        END IF

        DO 3170 J=1,12

          MONTH=0

          DO 3150 K=1,6

            MONTH=MONTH+EXPECT(SITE1,J,K)
     +             +EXPECT(SITE2,J,K)

3150        CONTINUE

          OBJ1(X)=OBJ1(X)+MONTH
          MEAN=MONTH/6
          SUMDEV=0

          DO 3160 K=1,6

            SUMDEV=SUMDEV+((EXPECT(SITE1,J,K)
     +             +EXPECT(SITE2,J,K)-MEAN)**2)

3160        CONTINUE

          OBJ2(X)=OBJ2(X)+(SUMDEV/6)

3170      CONTINUE
3180      CONTINUE
3190    CONTINUE
3200  CONTINUE


C
C
C   Perform computations for p=3 alternatives
C
C
    X=78
    DO 3300 SITE1=1,10
      DO 3290 SITE2=(SITE1+1),11
        DO 3280 SITE3=(SITE2+1),12
```

```
          X=X+1
          OBJ1(X)=0
          OBJ2(X)=0

          IF (.NOT.FEASIB(X)) THEN

            OBJ2(X)=1.0D20
            GO TO 3270

          END IF

          DO 3260 J=1,12

            MONTH=0

            DO 3240 K=1,6

              MONTH=MONTH+EXPECT(SITE1,J,K)
     +           +EXPECT(SITE2,J,K)+EXPECT(SITE3,J,K)

3240          CONTINUE

            OBJ1(X)=OBJ1(X)+MONTH
            MEAN=MONTH/6
            SUMDEV=0

            DO 3250 K=1,6

              SUMDEV=SUMDEV+((EXPECT(SITE1,J,K)
     +             +EXPECT(SITE2,J,K)
     +             +EXPECT(SITE3,J,K)-MEAN)**2)

3250          CONTINUE

            OBJ2(X)=OBJ2(X)+(SUMDEV/6)

3260        CONTINUE
3270        CONTINUE
3280      CONTINUE
3290    CONTINUE
3300  CONTINUE


C     Print OBJ1(X) and OBJ2(X) to file

      OPEN(UNIT=92,FILE='OBJFCN.OUT',STATUS='NEW')
      WRITE(92,'(1X,A)') 'OBJECTIVE FUNCTION VALUES'
      WRITE(92,3301) 'ALT NO','OBJ1','OBJ2'
3301  FORMAT(1X,A,6X,A,9X,A)

      DO 3303 X=1,298
        WRITE(92,3302) (X,OBJ1(X),OBJ2(X))
3302    FORMAT(1X,I4,3X,D12.6,1X,D12.6)
```

```
3303 CONTINUE

   CLOSE(92)

   RETURN

   END
*************************************************************
*              SUBPROGRAM EFFSET
*************************************************************
*
* Purpose  : Determine which alternatives are
*          non-dominated
*
* Variables : - Reals -
*             EXPECT: expected value (number of blocks)
*             OBJ1  : value of first objective function
*             OBJ2  : value of second objective func
*                   tion
*           - Integers -
*             I    : loop counter (location number)
*             J    :  "     "   (month number)
*             K    :  "     "   (satellite number)
*             X    : alternative number
*             EFF1 : no. of alt in N-set p=1
*             EFF2 : no. of alt in N-set p=2
*             EFF3 : no. of alt in N-set p=3
*           - Logicals -
*             FEASIB: feasibility record
*             NSET  : N-point record
*
*************************************************************

   SUBROUTINE EFFSET(FEASIB,OBJ1,OBJ2,NSET,
  +              EFF1,EFF2,EFF3)

   REAL*8 OBJ1(298),OBJ2(298)

   INTEGER I,J,K,X,EFF1,EFF2,EFF3

   LOGICAL FEASIB(298),NSET(298)

C
C
C  Initialize the NSET vector
C
C

   DO 4100 X=1,298
     NSET(X)=.TRUE.
4100 CONTINUE

C
```

```
C
C   Determine NSET for p=1 problem
C
C

    DO 4200 X=1,12

      IF (.NOT.FEASIB(X)) THEN
        NSET(X)=.FALSE.
        GO TO 4190
      END IF



      I=1

*------------------------------------------------------------
* Repeat Until Structure Begins Here

4180    CONTINUE

*       Check dominance against all other feasible alterna
*       tives

      IF((I.NE.X).AND.(FEASIB(I))) THEN

        IF((OBJ1(I).GE.OBJ1(X)).AND.
   +     (OBJ2(I).LE.OBJ2(X)))
   +     NSET(X)=.FALSE.

      END IF

      I=I+1

      IF(NSET(X).AND.(I.LE.12)) GO TO 4180

* Repeat Until Structure Ends Here... Next Statement is
* reached when either NSET(X)=.FALSE. or when I=13
*------------------------------------------------------------

4190    CONTINUE
4200  CONTINUE


C
C
C   Determine NSET for p=2 problem
C
C

    DO 4300 X=13,78

      IF (.NOT.FEASIB(X)) THEN
```

```
        NSET(X)=.FALSE.
        GO TO 4290
      END IF


      I=13

*-------------------------------------------------------
* Repeat Until Structure Begins Here

4280    CONTINUE

*       Check dominance against all other feasible alterna
*       tives

      IF((I.NE.X).AND.(FEASIB(I))) THEN

        IF((OBJ1(I).GE.OBJ1(X)).AND.
   +     (OBJ2(I).LE.OBJ2(X)))
   +     NSET(X)=.FALSE.

      END IF

      I=I+1

      IF(NSET(X).AND.(I.LE.78)) GO TO 4280

* Repeat Until Structure Ends Here... Next Statement is
* reached when either NSET(X)=.FALSE. or when I=79
*-------------------------------------------------------

4290    CONTINUE
4300  CONTINUE


C
C
C    Determine NSET for p=3 problem
C
C

     DO 4400 X=79,298

      IF (.NOT.FEASIB(X)) THEN
        NSET(X)=.FALSE.
        GO TO 4390
      END IF


      I=79

*-------------------------------------------------------
* Repeat Until Structure Begins Here
```

```
4380   CONTINUE

*      Check dominance against all other feasible alterna
*      tives

       IF((I.NE.X).AND.(FEASIB(I))) THEN

         IF((OBJ1(I).GE.OBJ1(X)).AND.
    +     (OBJ2(I).LE.OBJ2(X)))
    +     NSET(X)=.FALSE.

       END IF

       I=I+1

       IF(NSET(X).AND.(I.LE.298)) GO TO 4380

* Repeat Until Structure Ends Here... Next Statement is
* reached when either NSET(X)=.FALSE. or when I=299
*-----------------------------------------------------

4390   CONTINUE
4400  CONTINUE

C
C
C   Print NSET(X) to file
C
C
     OPEN(UNIT=94,FILE='EFFSET.OUT',STATUS='NEW')
     WRITE(94,*)
     WRITE(94,'(1X,A)') 'EFFICIENT SET'

     DO 4405 J=1,295,5
       WRITE(94,'(5(3X,I5,L3))')((X,NSET(X)),X=J,J+4)
4405  CONTINUE
     WRITE(94,'(3(3X,I5,L3))')((X,NSET(X)),X=296,298)

     CLOSE(94)


C
C
C   Count number of efficient alternatives
C
C
     EFF1=0
     EFF2=0
     EFF3=0

     DO 4500 X=1,12
       IF (NSET(X)) EFF1=EFF1+1
4500  CONTINUE
```

```
   DO 4600 X=13,78
     IF (NSET(X)) EFF2=EFF2+1
4600 CONTINUE

   DO 4700 X=79,298
     IF (NSET(X)) EFF3=EFF3+1
4700 CONTINUE


   RETURN

   END

*************************************************************
*              SUBPROGRAM IDLCAL
*************************************************************
*
* Purpose  : Determine ideal solutions
*
* Variables : - Reals -
*           OBJ1  : value of first  objective func
*                 tion
*           OBJ2  : value of second objective func
*                 tion
*           P1OBJ1: value of first  objective (p=1)
*           P1OBJ2: value of second objective (p=1)
*           P2OBJ1: value of first  objective (p=2)
*           P2OBJ2: value of second objective (p=2)
*           P3OBJ1: value of first  objective (p=3)
*           P3OBJ2: value of second objective (p=3)
*           IDLP1 : ideal solutions (p=1)
*           IDLP2 : ideal solutions (p=2)
*           IDLP3 : ideal solutions (p=3)
*         - Integers -
*           I     : loop counter
*           PERM? : vectors required by RSORT
*
* Subprograms:
*         RSORT - subroutine to sort real array
*              by algebraic value
*
*************************************************************

   SUBROUTINE IDLCAL(OBJ1,OBJ2,IDLP1,IDLP2,IDLP3)

   REAL*8 OBJ1(298),OBJ2(298),IDLP1(2),IDLP2(2),IDLP3(2)

   REAL*8 P1OBJ1(12),P1OBJ2(12),P2OBJ1(66),P2OBJ2(66)

   REAL*8 P3OBJ1(220),P3OBJ2(220)

   INTEGER I, PERM1(12),PERM2(66),PERM3(220)
C
```

```
C
C    Extract values for P?OBJ? vectors from the
C    OBJ1 and OBJ2 vectors
C
C
     DO 5000 I=1,12
       P1OBJ1(I)=OBJ1(I)
       P1OBJ2(I)=OBJ2(I)*(-1.)
5000 CONTINUE

     DO 5100 I=1,66
       P2OBJ1(I)=OBJ1(I+12)
       P2OBJ2(I)=OBJ2(I+12)*(-1.)
5100 CONTINUE

     DO 5200 I=1,220
       P3OBJ1(I)=OBJ1(I+78)
       P3OBJ2(I)=OBJ2(I+78)*(-1.)
5200 CONTINUE



     CALL RSORT(12,P1OBJ1,PERM1)
     CALL RSORT(12,P1OBJ2,PERM1)
     CALL RSORT(66,P2OBJ1,PERM2)
     CALL RSORT(66,P2OBJ2,PERM2)
     CALL RSORT(220,P3OBJ1,PERM3)
     CALL RSORT(220,P3OBJ2,PERM3)

     IDLP1(1)=P1OBJ1(12)
     IDLP1(2)=P1OBJ2(12)*(-1.)
     IDLP2(1)=P2OBJ1(66)
     IDLP2(2)=P2OBJ2(66)*(-1.)
     IDLP3(1)=P3OBJ1(220)
     IDLP3(2)=P3OBJ2(220)*(-1.)

     RETURN

     END

*************************************************************
*               SUBPROGRAM UTILS
*************************************************************
*
* Purpose   : Convert OBJ1 and OBJ2 to utils
*
* Variables :  - Reals -
*              OBJ1  :  value of first objective function
*              OBJ2  :  value of second objective func
*                         tion
*              UTIL1 :  objective 1 in utils
*              UTIL2 :  objective 2 in utils
*              LOW   :  lowest value of objective ?
```

```
*           HIGH  : highest value of objective ?
*           RANGE : range of values
*         - Integers -
*           I     : loop counter
*           PERM? : permutation vectors
*         - Logicals -
*           FEASIB: feasibility record
*           FOUND : loop stopper
*
**********************************************************

      SUBROUTINE UTILS(OBJ1,OBJ2,FEASIB)

      REAL*8 OBJ1(298),OBJ2(298)

      REAL*8 LOW,HIGH,RANGE

      LOGICAL FEASIB(298),FOUND

      REAL*8 F1P1(12),F1P2(66),F1P3(220)

      REAL*8 F2P1(12),F2P2(66),F2P3(220)

      INTEGER I,PERM1(12),PERM2(66),PERM3(220)


C
C
C   Load F?P? vectors (obj2 vectors multiplied by
C   negative 1 because this objective is being
C   minimized)
C
C

      DO 5300 I=1,12
        F1P1(I)=OBJ1(I)
        F2P1(I)=OBJ2(I)*(-1.)
        PERM1(I)=I
5300  CONTINUE

      DO 5301 I=1,66
        F1P2(I)=OBJ1(I+12)
        F2P2(I)=OBJ2(I+12)*(-1.)
        PERM2(I)=I
5301  CONTINUE

      DO 5302 I=1,220
        F1P3(I)=OBJ1(I+78)
        F2P3(I)=OBJ2(I+78)*(-1.)
        PERM3(I)=I
5302  CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C                              C
C   First, convert the OBJ1 vector to utils   C
C                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

    CALL RSORT(12,F1P1,PERM1)
    CALL RSORT(66,F1P2,PERM2)
    CALL RSORT(220,F1P3,PERM3)


*p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1
C
C   Determine high and low values for objective
C   function 1 of the p=1 problem
C
C

*   If highest value in vector is not feasible,
*   then make following assignments and skip the
*   DO WHILE block (recall that OBJ1=0 for an
*   infeasible alternative)
*

    IF(.NOT.FEASIB(PERM1(12))) THEN

      HIGH=1.
      LOW =0.
      GO TO 5310

    END IF

*--------------------------------------------------------
*   DO WHILE structure begins here
*   Finds first feasible value in vector F1P1

    HIGH=F1P1(12)
    I=1
    FOUND=.FALSE.

 5305 CONTINUE

    IF (FEASIB(PERM1(I))) THEN
      LOW=OBJ1(PERM1(I))
      FOUND=.TRUE.
    ELSE
      I=I+1
    END IF

    IF ((.NOT.FOUND).AND.(I.LT.12)) THEN
      GO TO 5305
    END IF

*   Next statement is reached when LOW has been found
```

```
*    or when the ONLY feasible element in the vector
*    is the last element
*-------------------------------------------------------

     IF (.NOT.FOUND) LOW=0

5310 CONTINUE


*    Convert p=1 section of OBJ1 to utils (infeasible
*    alternatives remain at zero by virtue of previous
*    assignments; a lone feasible alternative
*    takes on a utility of 1)

     RANGE=HIGH-LOW

     DO 5315 I=1,12

       IF(FEASIB(I)) THEN
       OBJ1(I)=ABS(OBJ1(I)-LOW)/RANGE
       END IF

5315 CONTINUE

*p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2
C
C    Determine high and low values for objective
C    function 1 of the p=2 problem
C
C

*    If highest value in vector is not feasible,
*    then make following assignments and skip the
*    DO WHILE block (recall that OBJ1=0 for an
*    infeasible alternative)
*

     IF(.NOT.FEASIB(PERM2(66)+12)) THEN

       HIGH=1.
       LOW =0.
       GO TO 5325

     END IF

*-------------------------------------------------------
*    DO WHILE structure begins here
*    Finds first feasible value in vector F1P2

     HIGH=F1P2(66)
     I=1
     FOUND=.FALSE.
```

```
5320 CONTINUE

    IF (FEASIB(PERM2(I)+12)) THEN
      LOW=OBJ1(PERM2(I)+12)
      FOUND=.TRUE.
    ELSE
      I=I+1
    END IF

    IF ((.NOT.FOUND).AND.(I.LT.66)) THEN
      GO TO 5320
    END IF

*    Next statement is reached when LOW has been found
*    or when the ONLY feasible element in the vector
*    is the last element
*-------------------------------------------------------

    IF (.NOT.FOUND) LOW=0

5325 CONTINUE

*    Convert p=2 section of OBJ1 to utiis (infeasible
*    alternatives remain at zero by virtue of previous
*    assignments;  a lone feasible alternative
*    takes on a utility of 1)

    RANGE=HIGH-LOW

    DO 5330 I=1,66

      IF (FEASIB(I+12)) THEN
      OBJ1(I+12)=ABS(OBJ1(I+12)-LOW)/RANGE
      END IF

5330 CONTINUE


*p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3
C
C    Determine high and low values for objective
C    function 1 of the p=3 problem
C
C

*    If highest value in vector is not feasible,
*    then make following assignments and skip the
*    DO WHILE block (recall that OBJ1=0 for an
*    infeasible alternative)
*

    IF(.NOT.FEASIB(PERM3(220)+78)) THEN
```

```
      HIGH=1.
      LOW =0.
      GO TO 5340

    END IF

*------------------------------------------------------
*   DO WHILE structure begins here
*   Finds first feasible value in vector F1P3

    HIGH=F1P3(220)
    I=1
    FOUND=.FALSE.

5335  CONTINUE

    IF (FEASIB(PERM3(I)+78)) THEN
      LOW=OBJ1(PERM3(I)+78)
      FOUND=.TRUE.
    ELSE
      I=I+1
    END IF

    IF ((.NOT.FOUND).AND.(I.LT.220)) THEN
      GO TO 5335
    END IF

*   Next statement is reached when LOW has been found
*   or when the ONLY feasible element in the vector
*   is the last element
*------------------------------------------------------

    IF (.NOT.FOUND) LOW=0

5340  CONTINUE

*   Convert p=3 section of OBJ1 to utils (infeasible
*   alternatives remain at zero by virtue of previous
*   assignments;  a lone feasible alternative
*   takes on a utility of 1)

    RANGE=HIGH-LOW

    DO 5345 I=1,220

      IF (FEASIB(I+78)) THEN
      OBJ1(I+78)=ABS(OBJ1(I+78)-LOW)/RANGE
      END IF

5345  CONTINUE
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   Next, convert the OBJ2 vector to utils
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


    DO 5346 I=1,12
      PERM1(I)=I
5346  CONTINUE

    DO 5347 I=1,66
      PERM2(I)=I
5347  CONTINUE

    DO 5348 I=1,220
      PERM3(I)=I
5348  CONTINUE



    CALL RSORT(12,F2P1,PERM1)
    CALL RSORT(66,F2P2,PERM2)
    CALL RSORT(220,F2P3,PERM3)


*p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1=p=1
C
C   Determine high and low values for objective
C   function 2 of the p=1 problem
C
C

*   If highest value in vector is not feasible,
*   then make following assignments and skip the
*   DO WHILE block
*

    IF(.NOT.FEASIB(PERM1(12))) THEN

      HIGH=1.
      LOW =0.
      GO TO 5355

    END IF

*-------------------------------------------------------
*   DO WHILE structure begins here
*   Finds first feasible value in vector F2P1

    LOW=F2P1(12)*(-1.)
    I=1
    FOUND=.FALSE.
```

```
5350 CONTINUE

    IF (FEASIB(PERM1(I))) THEN
      HIGH=OBJ2(PERM1(I))
      FOUND=.TRUE.
    ELSE
      I=I+1
    END IF

    IF ((.NOT.FOUND).AND.(I.LT.12)) THEN
      GO TO 5350
    END IF

*   Next statement is reached when HIGH has been found
*   or when the ONLY feasible element in the vector
*   is the last element
*-------------------------------------------------------

    IF (.NOT.FOUND) HIGH=0

5355 CONTINUE


*   Convert p=1 section of OBJ2 to utils (infeasible
*   alternatives are set to zero); a lone feasible
*   alternative takes on a utility of 1

    RANGE=ABS(HIGH-LOW)

    DO 5360 I=1,12

      IF(.NOT.FEASIB(I)) THEN
       OBJ2(I)=0.

      ELSE
        OBJ2(I)=ABS(HIGH - OBJ2(I))/RANGE

      END IF

5360 CONTINUE

*p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2=p=2
C
C   Determine high and low values for objective
C   function 2 of the p=2 problem
C
C

*   If highest value in vector is not feasible,
*   then make following assignments and skip the
*   DO WHILE block
*
```

```
      IF(.NOT.FEASIB(PERM2(66)+12)) THEN

        HIGH=1.
        LOW =0.
        GO TO 5370

      END IF

*-------------------------------------------------------
*     DO WHILE structure begins here
*     Finds first feasible value in vector F2P2

      LOW=F2P2(66)*(-1.)
      I=1
      FOUND=.FALSE.

 5365 CONTINUE

      IF (FEASIB(PERM2(I)+12)) THEN
        HIGH=OBJ2(PERM2(I)+12)
        FOUND=.TRUE.
      ELSE
        I=I+1
      END IF

      IF ((.NOT.FOUND).AND.(I.LT.66)) THEN
        GO TO 5365
      END IF

*     Next statement is reached when HIGH has been found
*     or when the ONLY feasible element in the vector
*     is the last element
*-------------------------------------------------------

      IF (.NOT.FOUND) HIGH=0

 5370 CONTINUE

*     Convert p=2 section of OBJ2 to utils (infeasible
*     alternatives are set to zero; a lone feasible alternative
*     takes on a utility of 1)

      RANGE=ABS(HIGH-LOW)


      DO 5375 I=1,66

        IF(.NOT.FEASIB(I+12)) THEN
          OBJ2(I+12)=0.

        ELSE
          OBJ2(I+12)=ABS(HIGH-OBJ2(I+12))/RANGE
```

```
      END IF

5375  CONTINUE


*p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3=p=3
C
C    Determine high and low values for objective
C    function 2 of the p=3 problem
C
C

*    If highest value in vector is not feasible,
*    then make following assignments and skip the
*    DO WHILE block
*

     IF(.NOT.FEASIB(PERM3(220)+78)) THEN

       HIGH=1.
       LOW =0.
       GO TO 5385

     END IF

*-------------------------------------------------------
*    DO WHILE structure begins here
*    Finds first feasible value in vector F2P3

     LOW=F2P3(220)*(-1.)
     I=1
     FOUND=.FALSE.

5380  CONTINUE

     IF (FEASIB(PERM3(I)+78)) THEN
       HIGH=OBJ2(PERM3(I)+78)
       FOUND=.TRUE.
     ELSE
       I=I+1
     END IF

     IF ((.NOT.FOUND).AND.(I.LT.220)) THEN
       GO TO 5380
     END IF

*    Next statement is reached when LOW has been found
*    or when the ONLY feasible element in the vector
*    is the last element
*-------------------------------------------------------

     IF (.NOT.FOUND) HIGH=0
```

```
5385 CONTINUE

*    Convert p=3 section of OBJ2 to utils (infeasible
*    alternatives are set to zero; a lone feasible
*    alternative takes on a utility of 1)

     RANGE=ABS(HIGH-LOW)


     DO 5390 I=1,220

       IF(.NOT.FEASIB(I+78)) THEN
         OBJ2(I+78)=0.
       ELSE
         OBJ2(I+78)=ABS(HIGH-OBJ2(I+78))/RANGE
       END IF

5390 CONTINUE

     OPEN (UNIT=96,FILE='UTILS.OUT',STATUS='NEW')


     WRITE(96,'(1X,A)') 'OBJ FUNCTION VALUES (UTILS)'
     WRITE(96,5393) 'ALT NO','OBJ1','OBJ2'
5393 FORMAT (1X,A,7X,A,7X,A)
     WRITE(96,*)

     DO 5395 I=1,298

       WRITE(96,5394) I,OBJ1(I),OBJ2(I)
5394     FORMAT(1X,I6,2X,D12.6,2X,D12.6)

5395 CONTINUE

     CLOSE(96)

     RETURN

     END


*************************************************************
*                SUBPROGRAM DEVCAL
*************************************************************
*
* Purpose   : Compute deviation from the ideal
*
* Variables : - Reals -
*             OBJ1  : value of first objective function
*             OBJ2  : value of second objective func
*                     tion
*             ONEDEV: deviations from ideal (p=1)
*             TWODEV: deviations from ideal (p=2)
```

```
*           TREDEV: deviations from ideal (p=3)
*         - Integers -
*           I    : loop counter
*           X    : alternative number
*         - Logicals -
*           FEASIB: feasibility record
*
************************************************************

      SUBROUTINE DEVCAL(OBJ1,OBJ2,FEASIB,
     +                  ONEDEV,TWODEV,TREDEV)

      REAL*8 OBJ1(298),OBJ2(298)

      REAL*8 ONEDEV(12),TWODEV(66),TREDEV(220)

      LOGICAL FEASIB(298)

      INTEGER I,X
C
C
C     Compute deviations for p=1 alternatives
C
C
      DO 6000 X=1,12

        IF (FEASIB(X)) THEN
          ONEDEV(X)=ABS(OBJ1(X)-1)
     +              +ABS(OBJ2(X)-1)
        ELSE
          ONEDEV(X)=5.5555555555

        END IF

6000  CONTINUE

C
C
C     Compute deviations for p=2 alternatives
C
C

      DO 6100 I=1,66

        X=I+12

        IF (FEASIB(X)) THEN
          TWODEV(I)=ABS(OBJ1(X)-1)
     +              +ABS(OBJ2(X)-1)
        ELSE
          TWODEV(I)=5.5555555555

        END IF
```

```
6100  CONTINUE

C
C
C    Compute deviations for p=3 alternatives
C
C

      DO 6200 I=1,220

        X=I+78

        IF (FEASIB(X)) THEN
          TREDEV(I)=ABS(OBJ1(X)-1)
     +              +ABS(OBJ2(X)-1)
        ELSE
          TREDEV(I)=5.5555555555

        END IF

6200  CONTINUE

C
C
C    Print ONEDEV, TWODEV, and TREDEV to file
C
C
      OPEN(UNIT=95,FILE='DEVIAT.OUT',STATUS='NEW')

      WRITE(95,*)
      WRITE(95,'(1X,A)')'DEVIATION FROM IDEAL (IN UTILS)'
      WRITE(95,*)
      WRITE(95,6201) 'ALT NO', 'DEVIATION'
6201  FORMAT(1X,A,6X,A)
      WRITE(95,*)

      DO 6202 I=1,12
        WRITE(95,6205) I, ONEDEV(I)
6202  CONTINUE

      DO 6203 I=1,66
        WRITE(95,6205) I+12, TWODEV(I)
6203  CONTINUE

      DO 6204 I=1,220
        WRITE(95,6205) I+78, TREDEV(I)
6204  CONTINUE

6205  FORMAT(1X,I4,8X,D12.6)

      CLOSE(95)

      RETURN
```

```
      END
***************************************************************
*                SUBPROGRAM PRIORI
***************************************************************
*
* Purpose  : Rank alternatives based on deviation from
*          ideal
*
* Variables : - Reals -
*             ONEDEV: deviation from ideal (p=1)
*             TWODEV: deviation from ideal (p=2)
*             TREDEV: deviation from ideal (p=3)
*          - Integers -
*             I     : loop counter
*             ONEPRM: permutation vector (p=1)
*             TWOPRM: permutation vector (p=2)
*             TREPRM: permutation vector (p=3)
*
* Subprograms:
*          RSORT  - subroutine to sort real array
*                by algebraic value and return the
*                permutations. Permutations are
*                return in vectors ONEPRM, TWOPRM,
*                and TREPRM.
*
***************************************************************

      SUBROUTINE PRIORI(ONEDEV,TWODEV,TREDEV,
     +            ONEPRM,TWOPRM,TREPRM)

      REAL*8 ONEDEV(12),TWODEV(66),TREDEV(220)

      INTEGER ONEPRM(12),TWOPRM(66),TREPRM(220)

C
C
C   Initialize the permutation vectors
C
C
      DO 7000 I=i,12
        ONEPRM(I)=I
7000  CONTINUE

      DO 7100 I=1,66
        TWOPRM(I)=I
7100  CONTINUE

      DO 7200 I=1,220
        TREPRM(I)=I
7200  CONTINUE

      CALL RSORT (12,ONEDEV,ONEPRM)
```

```
      CALL RSORT (66,TWODEV,TWOPRM)

      CALL RSORT (220,TREDEV,TREPRM)

      RETURN

      END
************************************************************
*                 SUBPROGRAM PRTOUT
************************************************************
*
* Purpose   : Print out results
*
* Variables : - Reals -
*             EXPECT: expected value (number of blocks)
*             OBJ1  : value of first objective function
*             OBJ2  : value of second objective func
*                     tion
*             ONEDEV: deviation from ideal (p=1)
*             TWODEV: deviation from ideal (p=2)
*             TREDEV: deviation from ideal (p=3)
*             IDLP1 : y-space coord of ideal (p=1)
*             IDLP2 : y-space coord of ideal (p=2)
*             IDLP3 : y-space coord of ideal (p=3)
*
*             Note: vectors of deviation from ideal
*                   have previously been sorted in
*                   ascending order. The ???PRM integer
*                   vectors are the associated permutation vectors.
*
*           - Integers -
*             I     : loop counter
*             J     :  "     "
*             K     :  "     "
*             X     : alternative number
*             FEA?  : number of feasibles in p=?
*             EFF?  : number of efficients in p=?
*
************************************************************

      SUBROUTINE PRTOUT (EXPECT,OBJ1,OBJ2,
     + ONEDEV,TWODEV,TREDEV,
     + ONEPRM,TWOPRM,TREPRM,IDLP1,IDLP2,IDLP3,
     + FEA1,FEA2,FEA3,EFF1,EFF2,EFF3)

      REAL*8 EXPECT(12,12,6),OBJ1(298),OBJ2(298)

      REAL*8 ONEDEV(12),TWODEV(66),TREDEV(220)

      REAL*8 IDLP1(2),IDLP2(2),IDLP3(2)

      INTEGER ONEPRM(12),TWOPRM(66),TREPRM(220)
```

```
      INTEGER I,J,K,X

      INTEGER FEA1,FEA2,FEA3,EFF1,EFF2,EFF3

      OPEN(UNIT=80,FILE='RESULTS.OUT',STATUS='NEW')
C
C
C     Print p=1 alternatives in ascending order of deviation
C     from the ideal
C
C
        WRITE(80,*)
        WRITE(80,*)
        WRITE(80,'(1X,A)') 'TOP 12 ALTERNATIVES (P=1)'
        WRITE(80,*)
        WRITE(80,8440)  'NUMBER FEASIBLE =',FEA1,' OF 12'
        WRITE(80,8441)  'NUMBER IN N-SET =',EFF1
        WRITE(80,8450) 'IDEAL OBJ1= ',IDLP1(1)
        WRITE(80,8450) 'IDEAL OBJ2= ',IDLP1(2)
        WRITE(80,*)
        WRITE(80,8500) 'ALT NO','OBJ1','OBJ2','DEVIATION'

      DO 8200 I=1,12

        WRITE(80,8600) ONEPRM(I),OBJ1(ONEPRM(I)),
     +             OBJ2(ONEPRM(I)),ONEDEV(I)

8200  CONTINUE
C
C
C     Print p=2 alternatives in ascending order of deviation
C     from the ideal (TOP 12)
C
C
        WRITE(80,*)
        WRITE(80,*)
        WRITE(80,'(1X,A)') 'TOP 12 ALTERNATIVES (P=2)'
        WRITE(80,*)
        WRITE(80,8440) 'NUMBER FEASIBLE= ',FEA2, ' OF 66'
        WRITE(80,8441) 'NUMBER IN N-SET= ',EFF2
        WRITE(80,8450) 'IDEAL OBJ1= ',IDLP2(1)
        WRITE(80,8450) 'IDEAL OBJ2= ',IDLP2(2)
        WRITE(80,*)
        WRITE(80,8500) 'ALT NO','OBJ1','OBJ2','DEVIATION'

      DO 8300 I=1,12

        WRITE(80,8600) TWOPRM(I)+12,
     +             OBJ1(TWOPRM(I)+12),
     +             OBJ2(TWOPRM(I)+12),
     +             TWODEV(I)

8300  CONTINUE
```

```
C
C
C   Print p=3 alternatives in ascending order of deviation
C   from the ideal (TOP 12)
C
C
      WRITE(80,*)
      WRITE(80,*)
      WRITE(80,'(1X,A)') 'TOP 12 ALTERNATIVES (P=3)'
      WRITE(80,*)
      WRITE(80,8440) 'NUMBER FEASIBLE= ',FEA3, ' OF 220'
      WRITE(80,8441) 'NUMBER IN N-SET= ',EFF3
      WRITE(80,8450) 'IDEAL OBJ1= ',IDLP3(1)
      WRITE(80,8450) 'IDEAL OBJ2= ',IDLP3(2)
      WRITE(80,*)
      WRITE(80,8500) 'ALT NO','OBJ1','OBJ2','DEVIATION'

      DO 8400 I=1,12

        WRITE(80,8600) TREPRM(I)+78,
     +              OBJ1(TREPRM(I)+78),
     +              OBJ2(TREPRM(I)+78),
     +              TREDEV(I)

8400  CONTINUE

8440    FORMAT(1X,A,I5,A)
8441    FORMAT(1X,A,I5)
8450    FORMAT(1X,A,1X,D12.7)
8500    FORMAT(1X,A,9X,A,11X,A,8X,A)
8600    FORMAT(2X,I3,3X,3(3X,F12.10))

C
C
C   Print expected value of number of useable blocks
C
C
      WRITE(80,*)
      WRITE(80,*)
      WRITE(80,*)

      DO 8650 K=1,6

        WRITE(80,'(1X,A,I2)') 'EXPECTED NUMBER OF OBS FOR SAT',K
        WRITE(80,'(1X,A)') 'LOCATIONS (ROWS) MONTHS (COLS)'
        WRITE(80,*)

        DO 8640 I=1,12

          WRITE(80,'(12(F6.0))')(EXPECT(I,J,K),J=1,12)


8640    CONTINUE
```

```
      WRITE(80,*)

8650  CONTINUE

C
C
C    Print legend of alternative numbers
C
C
      OPEN(UNIT=81,FILE='ALTLST.OUT',STATUS='NEW')

      WRITE(81,*)
      WRITE(81,'(1X,A)') 'ALTERNATIVE NUMBER LEGEND (P=1)'
      WRITE(81,*)
      WRITE(81,'(1X,A)') 'ALT NO  SITE NO'

      DO 8700 X=1,12

        WRITE(81,'(1X,I3,I8)') X,X

8700  CONTINUE

      WRITE(81,*)
      WRITE(81,'(1X,A)') 'ALTERNATIVE NUMBER LEGEND (P=2)'
      WRITE(81,*)
      WRITE(81,'(1X,A)') 'ALT NO  SITE NO'

      X=12
      DO 8800 I=1,11
        DO 8790 J=I+1,12

          X=X+1
          WRITE(81,8780) X,I,'-',J
8780      FORMAT(3X,I3,2X,I2,A,I2)

8790    CONTINUE
8800  CONTINUE

      WRITE(81,*)
      WRITE(81,'(1X,A)') 'ALTERNATIVE NUMBER LEGEND (P=3)'
      WRITE(81,*)
      WRITE(81,'(1X,A)') 'ALT NO  SITE NO'

      X=78
      DO 8900 I=1,10
        DO 8890 J=I+1,11
          DO 8880 K=J+1,12

            X=X+1
            WRITE(81,8870) X,I,'-',J,'-',K
8870        FORMAT(3X,I3,2X,I2,A,I2,A,I2)

8880      CONTINUE
```

```
8890    CONTINUE
8900  CONTINUE

    CLOSE(80)
    CLOSE(81)

    RETURN

    END
************************************************************
*                SUBPROGRAM RSORT
************************************************************
*
* Purpose  : Sort real array in ascending order and
*           return sorted array along with permutation
*           vector
*
* Variables : - Reals -
*             RARRAY: array to be sorted
*           - Integers -
*             I,J   : loop counters
*             N     : number of elements
*             LOW   : index number of lowest value
*             LIMIT : maximum size of RARRAY
*
* Subprograms:
*           RSWAP - swap two pairs of values
*
* Reference : This subroutine is an adaptation of a
*           selection sort subroutine presented by:
*           McCracken D. and Salmon W. "Computing for
*           Engineers and Scientists with FORTRAN 77".
*           New York: John Wiley & Sons Inc, 1988.
*
************************************************************

    SUBROUTINE RSORT(N,RARRAY,PERM)

    INTEGER I,J,N,LOW

    INTEGER PERM(N)

    REAL*8 RARRAY(N)


    DO 9100 I=1,N-1

C      -- Find LOW, the remaining lowest value
C         in unsorted part of the array

      LOW=I
      DO 9000 J=(I+1),N
```

```
          IF(RARRAY(J).LT.RARRAY(LOW)) LOW=J

9000      CONTINUE

C        -- Swap lowest element found with element
C           at the beginning of the unsorted part
C           of the array

        CALL RSWAP(RARRAY(I),RARRAY(LOW),PERM(I),PERM(LOW))

9100  CONTINUE

      RETURN

      END

************************************************************
*                SUBPROGRAM RSWAP
************************************************************
*
* Purpose  : Swap two real values and two integer values
*
* Variables : - Reals -
*              X,Y    : values to be swapped
*              RTEMP : temporary storage
*            - Integers -
*              I,J   : values to be swapped
*              TEMP  : temporary storage
*
************************************************************

      SUBROUTINE RSWAP(X,Y,I,J)

      INTEGER I,J,TEMP

      REAL*8 X,Y,RTEMP

      TEMP=I
      I=J
      J=TEMP

      RTEMP=X
      X=Y
      Y=RTEMP

      RETURN

      END
```

<PROBA.DAT>

```
---------------------------------------------
 61  54  46  37  28  22  24  33  42  51  59  63
 64  55  46  34  22  12  16  28  40  51  61  66
 59  53  46  38  31  26  28  35  43  50  57  61
 58  53  46  39  32  28  29  36  43  50  57  60
 58  53  46  39  32  28  29  36  43  50  57  60
100  85  33 000 000 000 000 000  17  61 100 100
 68  57  45  30  11 000 000  23  37  52  64  71
 73  59  44  24 000 000 000  12  27  53  68  78
 57  52  47  40  34  31  32  38  44  50  55  58
 60  53  46  38  30  25  29  34  43  50  58  61
 65  56  45  32  18 000  11  27  40  53  62  68
 61  54  46  36  27  20  23  32  42  51  59  63
---------------------------------------------
```

<PROBB.DAT>

```
---------------------------------------------
 86  88  92  94  95  97  98  99  96  91  88  88
 89  92  93  94  96  97  97  95  91  87  88  90
 97  97  98  99 100 100 100 100 100  98  96  96
 96  96  96  97 100 100 100 100  99  99  97  96
 71  71  72  84  90  90  91  95  90  84  77  71
 97  97  98  00  00  00  00  00  95  97  97  96
 93  90  96  97  97  00  00  98  97  95  93  95
 99  99 100 100  00  00  00  99 100 100 100  99
 95  97  97  97  99 100 100 100 100  99  97  96
 96  97  97  98  98  98  99  99  98  98  98  97
 97  97  99  99  99  00 100 100  99  97  97  96
100 100 100 100  99  99  99 100 100  99  99  99
---------------------------------------------
```

<PROBC.DAT>

```
---------------------------------------------
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100
---------------------------------------------
```

<PROBD.DAT>

```
------------------------
 28 000 000 100 100 100
 30 000 100 100 100 000
 27 000 100 100 100 100
 27 100 100 100 000 000
 27 100 100 100 000 000
 33 000 000 000 000 000
 31 100 100 000 000 000
 32 000 000 000 000 000
 26 100 100 100 100 000
 26 000 100 100 100 100
 31 000 000 100 100 100
 29 000 100 100 100 100
------------------------
```

<PROBE.DAT>

```
------------------------------------------------
SATELLITE 1
 28 31 38 37 32 31 28 38 37 33 34 30
 47 47 50 35 27 29 35 40 24 20 29 42
 14 21 35 39 38 45 49 52 46 42 19 15
 39 39 36 34 33 41 41 45 43 40 34 36
 21 19 27 26 22 26 29 29 32 25 23 20
 51 49 50 00 00 00 00 00 21 37 45 54
 44 46 43 39 16 00 00 19 21 19 30 44
 43 43 51 42 00 00 00 31 23 22 37 42
 20 27 32 36 39 47 50 47 46 36 23 19
 42 40 41 49 51 52 54 58 46 48 41 38
 35 38 36 43 36 00 28 33 34 24 23 31
 35 35 34 34 31 31 35 38 32 36 28 31
SATELLITE 2
 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00
 44 42 41 42 40 47 52 51 49 46 41 43
 27 23 31 33 27 31 39 32 35 31 29 25
 00 00 00 00 00 00 00 00 00 00 00 00
 48 47 45 45 19 00 00 20 24 21 34 49
 00 00 00 00 00 00 00 00 00 00 00 00
 25 30 36 45 45 54 60 52 52 42 28 24
 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00
SATELLITE 3
 00 00 00 00 00 00 00 00 00 00 00 00
 51 50 54 43 31 34 45 42 28 23 33 48
 17 23 37 46 42 49 59 56 50 45 23 17
 43 42 41 43 39 47 52 51 49 45 42 42
 27 23 30 34 27 31 39 32 37 30 30 25
 00 00 00 00 00 00 00 00 00 00 00 00
```

47 48 45 45 20 00 00 20 24 21 34 49
00 00 00 00 00 00 00 00 00 00 00 00
25 31 37 46 45 55 62 53 52 42 28 25
44 39 37 46 45 55 60 60 50 50 43 41
00 00 00 00 00 00 00 00 00 00 00 00
38 38 37 42 36 36 44 42 37 42 32 36
SATELLITE 4
31 33 41 44 33 34 33 39 40 36 36 35
51 49 55 44 32 33 44 44 28 23 34 47
19 28 40 49 45 53 61 58 52 48 25 19
44 41 40 41 39 47 52 50 49 45 40 42
26 21 29 31 25 29 36 30 33 27 26 23
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
24 30 37 46 45 54 62 53 51 42 28 23
47 41 45 55 51 57 62 59 51 51 44 44
37 40 38 49 41 00 34 33 37 26 26 34
41 40 40 45 39 39 47 45 40 43 35 38
SATELLITE 5
33 37 43 45 35 36 35 40 41 37 37 35
51 50 54 43 31 33 44 42 28 22 34 47
20 26 41 48 47 53 61 60 54 50 26 21
40 40 37 40 37 43 49 47 45 43 37 39
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
25 33 37 45 45 53 60 51 50 41 27 24
45 40 45 55 51 56 62 60 52 50 45 44
39 42 40 52 41 00 36 35 39 29 27 36
41 40 40 45 39 39 47 45 40 43 35 38
SATELLITE 6
33 33 43 45 35 35 35 41 41 37 37 35
49 47 52 40 30 29 41 40 25 21 32 45
18 24 41 50 45 52 61 59 53 48 25 19
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
44 39 44 55 49 55 60 60 50 49 43 42
39 41 41 52 42 00 38 37 39 28 27 37
39 38 38 44 37 37 46 44 38 42 33 37
END-DATA

<PROBF.DAT>

```
-------------------------------------------------
SATELLITE 1
 100  99  97 100 100 100 100 100  89  99 100 100
 100 100 100 100 100 100 000 100  90  99 100 100
 100 100 100 100 100 100 100  99  86  99 100 100
 100  99  85 100 000 000 000 100  84  99 100 100
 100 100  82  98 000 000 000 000  92 100 100 100
 100 100 100 000 000 000 000 000 100 100 100 100
 100 100  98 100 100 000 000 000  96 100 100 100
 100 100 100 100 100 000 000 100  98 100 100 100
 100 100 100 000 000 000 100  96  82  99 100 100
 100 100 100 100 100 100 100  99  86  99 100 100
 100  99  98 100 100 000 100 100  94  99 100 100
 100 100 100 100 100 100 100 100  88  99 100 100
SATELLITE 2
 000 000 000 000 000 000 000 000 000 000 000 000
 000 000 000 000 000 000 000 000 000 000 000 000
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
 000 000 000 000 000 000 000 000 000 000 000 000
 000 000 000 000 000 000 000 000 000 000 000 000
 000 000 000 000 000 000 000 000 000 000 000 000
SATELLITE 3
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
SATELLITE 4
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
 000 000 000 000 000 000 000 000 000 000 000 000
 000 000 000 000 000 000 000 000 000 000 000 000
 000 000 000 000 000 000 000 000 000 000 000 000
 100 100  91  97 100 100 100 100  91  97 100 100
 100 100  91  97 100 100 100 100  91  97 100 100
```

100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
SATELLITE 5
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
SATELLITE 6
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
100 100 91 97 100 100 100 100 91 97 100 100
END-DATA

<OBSREQ.DAT>

----

30 30 30 30 30 30

<Output of RESULTS.OUT>

TOP 12 ALTERNATIVES (P=1)

NUMBER FEASIBLE = 0 OF 12
NUMBER IN N-SET = 0
IDEAL OBJ1= .0000000D+00
IDEAL OBJ2= .1000000D+21

ALT NO	OBJ1	OBJ2	DEVIATION
1	0.0000000000	0.0000000000	5.5555555555
2	0.0000000000	0.0000000000	5.5555555555
3	0.0000000000	0.0000000000	5.5555555555
4	0.0000000000	0.0000000000	5.5555555555
5	0.0000000000	0.0000000000	5.5555555555
6	0.0000000000	0.0000000000	5.5555555555
7	0.0000000000	0.0000000000	5.5555555555
8	0.0000000000	0.0000000000	5.5555555555
9	0.0000000000	0.0000000000	5.5555555555
10	0.0000000000	0.0000000000	5.5555555555
11	0.0000000000	0.0000000000	5.5555555555
12	0.0000000000	0.0000000000	5.5555555555

TOP 12 ALTERNATIVES (P=2)

NUMBER FEASIBLE= 12 OF 66
NUMBER IN N-SET= 10
IDEAL OBJ1= .1635424D+06
IDEAL OBJ2= .3182542D+07

ALT NO	OBJ1	OBJ2	DEVIATION
15	0.3573576935	0.7414240023	0.9012183041
55	0.5074134196	0.5487584640	0.9438281164
35	0.2653469046	0.7818326575	0.9528204379
57	0.3023568314	0.7348871726	0.9627559960
39	0.7579334850	0.2772144489	0.9648520661
20	0.4925865804	0.5323157265	0.9750976931
75	0.7949434118	0.2267222203	0.9783343678
48	0.8647711131	0.1387734001	0.9964554868
16	0.0000000000	1.0000000000	1.0000000000
73	1.0000000000	0.0000000000	1.0000000000
50	0.6597145250	0.3093213101	1.0309641649
34	0.6227045981	0.3459432993	1.0313521026

TOP 12 ALTERNATIVES (P=3)

NUMBER FEASIBLE= 90 OF 220
NUMBER IN N-SET= 23
IDEAL OBJ1= .2369832D+06
IDEAL OBJ2= .2479417D+07

ALT NO	OBJ1	OBJ2	DEVIATION
273	0.5524938430	0.7299858072	0.7175203497
103	0.7636153008	0.5073201559	0.7290645433
283	0.7251571578	0.5448532198	0.7299896223
185	0.6227261163	0.6425187066	0.7347551771
89	0.6328504747	0.6310528198	0.7360967055
240	0.7534909423	0.5083462058	0.7381628519
120	0.4510513917	0.8101480309	0.7388005774
289	0.5402020435	0.7182611271	0.7415368295
105	0.6528433128	0.6022687666	0.7448879206
225	0.4794429220	0.7729745650	0.7475825130
275	0.4417218551	0.7967710813	0.7615070637
242	0.6427189544	0.5957637864	0.7615172592

EXPECTED NUMBER OF OBS FOR SAT 1
LOCATIONS (ROWS) MONTHS (COLS)

367.	329.	390.	311.	213.	160.	165.	310.	321.	379.	427.	416.
717.	575.	573.	290.	153.	87.	0.	285.	204.	235.	404.	668.
193.	235.	380.	342.	284.	273.	331.	434.	397.	491.	243.	212.
523.	428.	326.	300.	0.	0.	0.	391.	359.	473.	439.	500.
208.	156.	177.	195.	0.	0.	0.	0.	266.	253.	235.	205.
1458.	1075.	476.	0.	0.	0.	0.	0.	97.	645.	1245.	1527.
770.	590.	504.	304.	47.	0.	0.	0.	194.	260.	478.	821.
888.	648.	641.	279.	0.	0.	0.	105.	168.	333.	696.	927.
251.	286.	339.	0.	0.	0.	371.	398.	373.	410.	276.	246.
562.	431.	425.	410.	348.	286.	360.	449.	374.	541.	524.	522.
611.	511.	435.	365.	178.	0.	85.	247.	339.	338.	370.	560.
553.	442.	405.	307.	215.	154.	206.	315.	296.	466.	410.	501.

EXPECTED NUMBER OF OBS FOR SAT 2
LOCATIONS (ROWS) MONTHS (COLS)

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2187.	1723.	1471.	1332.	1143.	1137.	1346.	1639.	1640.	1972.	1959.	2211.
993.	698.	834.	906.	694.	675.	919.	977.	1065.	1128.	1100.	951.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2710.	1944.	1579.	1097.	181.	0.	0.	402.	677.	898.	1748.	2951.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
1209.	1220.	1333.	1463.	1352.	1446.	1714.	1764.	1799.	1800.	1291.	1193.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

EXPECTED NUMBER OF OBS FOR SAT 3
LOCATIONS (ROWS) MONTHS (COLS)

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2594.	2040.	1877.	1152.	585.	342.	624.	997.	801.	884.	1531.	2546.
869.	954.	1355.	1450.	1162.	1101.	1475.	1750.	1690.	1910.	1087.	889.
2138.	1723.	1471.	1363.	1114.	1137.	1346.	1639.	1640.	1929.	2006.	2160.
993.	698.	807.	933.	694.	675.	919.	977.	1126.	1091.	1138.	951.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2654.	1986.	1579.	1097.	191.	0.	0.	402.	677.	898.	1748.	2951.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
1209.	1261.	1370.	1496.	1352.	1473.	1771.	1798.	1799.	1800.	1291.	1243.
2263.	1617.	1341.	1436.	1181.	1164.	1538.	1803.	1657.	2122.	2112.	2166.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2070.	1655.	1383.	1267.	859.	616.	894.	1200.	1222.	1836.	1615.	2005.

EXPECTED NUMBER OF OBS FOR SAT 4
LOCATIONS (ROWS) MONTHS (COLS)

1452.	1265.	1410.	1283.	784.	627.	693.	1138.	1268.	1447.	1615.	1732.
2594.	1999.	1912.	1179.	603.	332.	610.	1045.	801.	884.	1577.	2493.
971.	1161.	1465.	1545.	1245.	1191.	1525.	1812.	1758.	2037.	1182.	993.
2187.	1682.	1435.	1300.	1114.	1137.	1346.	1607.	1640.	1929.	1911.	2160.
956.	637.	780.	851.	643.	631.	848.	916.	1004.	982.	986.	875.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
1160.	1220.	1370.	1496.	1352.	1446.	1771.	1798.	1764.	1800.	1291.	1143.
2417.	1700.	1631.	1717.	1339.	1207.	1589.	1773.	1690.	2164.	2161.	2324.
2083.	1752.	1375.	1301.	652.	0.	334.	795.	1152.	1158.	1351.	1982.
2233.	1742.	1495.	1358.	931.	667.	955.	1286.	1321.	1880.	1766.	2116.

EXPECTED NUMBER OF OBS FOR SAT 5
LOCATIONS (ROWS) MONTHS (COLS)

1546.	1418.	1478.	1312.	831.	664.	735.	1167.	1300.	1487.	1660.	1732.
2594.	2040.	1877.	1152.	585.	332.	610.	997.	801.	845.	1577.	2493.
1022.	1078.	1502.	1513.	1301.	1191.	1525.	1875.	1826.	2122.	1229.	1098.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
1209.	1342.	1370.	1463.	1352.	1420.	1714.	1730.	1730.	1738.	1245.	1193.
2314.	1658.	1631.	1717.	1339.	1185.	1589.	1803.	1723.	2122.	2210.	2324.
2195.	1840.	1448.	1381.	652.	0.	354.	844.	1214.	1291.	1403.	2098.
2233.	1742.	1495.	1358.	931.	667.	955.	1286.	1321.	1880.	1766.	2116.

EXPECTED NUMBER OF OBS FOR SAT 6
LOCATIONS (ROWS) MONTHS (COLS)

1546.	1265.	1478.	1312.	831.	645.	735.	1196.	1300.	1487.	1660.	1732.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
920.	995.	1502.	1576.	1245.	1168.	1525.	1844.	1792.	2037.	1182.	993.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2263.	1617.	1595.	1717.	1286.	1164.	1538.	1803.	1657.	2079.	2112.	2219.
2195.	1796.	1484.	1381.	668.	0.	373.	892.	1214.	1247.	1403.	2156.
2124.	1655.	1420.	1328.	883.	633.	935.	1257.	1255.	1836.	1665.	2060.

<Output of FEASIB.OUT>

FEASIBILITY LIST

1 F	2 F	3 F	4 F	5 F
6 F	7 F	8 F	9 F	10 F
11 F	12 F	13 F	14 F	15 T
16 T	17 F	18 F	19 F	20 T
21 F	22 F	23 F	24 F	25 F
26 F	27 F	28 F	29 F	30 F
31 F	32 F	33 F	34 T	35 T
36 F	37 F	38 F	39 T	40 F
41 F	42 F	43 F	44 F	45 F
46 F	47 F	48 T	49 F	50 T
51 F	52 F	53 F	54 F	55 T
56 F	57 T	58 F	59 F	60 F
61 F	62 F	63 F	64 F	65 F
66 F	67 F	68 F	69 F	70 F
71 F	72 F	73 T	74 F	75 T

76 F	77 F	78 F	79 F	80 T
81 T	82 F	83 F	84 F	85 T
86 F	87 F	88 F	89 T	90 T
91 F	92 F	93 F	94 T	95 F
96 F	97 F	98 T	99 T	100 T
101 T	102 T	103 T	104 T	105 T
106 T	107 T	108 T	109 T	110 T
111 T	112 T	113 F	114 F	115 T
116 F	117 F	118 F	119 F	120 T
121 F	122 F	123 F	124 T	125 F
126 F	127 F	128 T	129 T	130 T
131 F	132 F	133 F	134 T	135 T
136 F	137 F	138 F	139 T	140 F
141 F	142 F	143 F	144 F	145 F
146 F	147 F	148 T	149 F	150 T
151 F	152 F	153 F	154 F	155 T
156 F	157 T	158 F	159 F	160 F
161 F	162 F	163 F	164 F	165 F
166 F	167 F	168 F	169 F	170 F
171 F	172 F	173 T	174 F	175 T
176 F	177 F	178 F	179 T	180 T
181 T	182 T	183 T	184 T	185 T
186 T	187 T	188 T	189 T	190 T
191 T	192 T	193 T	194 F	195 F
196 T	197 F	198 F	199 F	200 F
201 T	202 F	203 F	204 F	205 T
206 F	207 F	208 F	209 T	210 T
211 T	212 F	213 F	214 F	215 F
216 F	217 F	218 F	219 T	220 F
221 T	222 F	223 F	224 F	225 T
226 F	227 T	228 F	229 F	230 T
231 F	232 T	233 F	234 T	235 F
236 T	237 T	238 F	239 T	240 T
241 T	242 T	243 F	244 F	245 F
246 T	247 F	248 T	249 F	250 F
251 T	252 F	253 T	254 F	255 T
256 F	257 T	258 T	259 F	260 T
261 T	262 T	263 T	264 F	265 F
266 F	267 F	268 F	269 F	270 F
271 F	272 F	273 T	274 F	275 T
276 F	277 F	278 F	279 F	280 F
281 F	282 F	283 T	284 F	285 T
286 F	287 F	288 F	289 T	290 F
291 T	292 F	293 F	294 F	295 T
296 T	297 T	298 F		

<Output of EFFSET.OUT>

EFFICIENT SET

1 F	2 F	3 F	4 F	5 F
6 F	7 F	8 F	9 F	10 F
11 F	12 F	13 F	14 F	15 T
16 T	17 F	18 F	19 F	20 F
21 F	22 F	23 F	24 F	25 F
26 F	27 F	28 F	29 F	30 F
31 F	32 F	33 F	34 T	35 T
36 F	37 F	38 F	39 T	40 F
41 F	42 F	43 F	44 F	45 F
46 F	47 F	48 T	49 F	50 T
51 F	52 F	53 F	54 F	55 T
56 F	57 F	58 F	59 F	60 F
61 F	62 F	63 F	64 F	65 F
66 F	67 F	68 F	69 F	70 F
71 F	72 F	73 T	74 F	75 T
76 F	77 F	78 F	79 F	80 F
81 F	82 F	83 F	84 F	85 F
86 F	87 F	88 F	89 T	90 F
91 F	92 F	93 F	94 F	95 F
96 F	97 F	98 F	99 T	100 F
101 F	102 F	103 T	104 F	105 T
106 T	107 T	108 T	109 F	110 F
111 F	112 F	113 F	114 F	115 F
116 F	117 F	118 F	119 F	120 T
121 F	122 F	123 F	124 F	125 F
126 F	127 F	128 T	129 F	130 F
131 F	132 F	133 F	134 F	135 F
136 F	137 F	138 F	139 F	140 F
141 F	142 F	143 F	144 F	145 F
146 F	147 F	148 F	149 F	150 F
151 F	152 F	153 F	154 F	155 F
156 F	157 F	158 F	159 F	160 F
161 F	162 F	163 F	164 F	165 F
166 F	167 F	168 F	169 F	170 F
171 F	172 F	173 F	174 F	175 F
176 F	177 F	178 F	179 F	180 F
181 F	182 F	183 F	184 T	185 T
186 T	187 F	188 F	189 F	190 F
191 F	192 F	193 F	194 F	195 F
196 F	197 F	198 F	199 F	200 F
201 F	202 F	203 F	204 F	205 F
206 F	207 F	208 F	209 T	210 F
211 F	212 F	213 F	214 F	215 F
216 F	217 F	218 F	219 F	220 F
221 F	222 F	223 F	224 F	225 T
226 F	227 T	228 F	229 F	230 F
231 F	232 F	233 F	234 F	235 F
236 F	237 T	238 F	239 F	240 T
241 T	242 F	243 F	244 F	245 F

246 T	247 F	248 T	249 F	250 F
251 F	252 F	253 F	254 F	255 F
256 F	257 F	258 F	259 F	260 F
261 F	262 F	263 F	264 F	265 F
266 F	267 F	268 F	269 F	270 F
271 F	272 F	273 T	274 F	275 F
276 F	277 F	278 F	279 F	280 F
281 F	282 F	283 T	284 F	285 F
286 F	287 F	288 F	289 F	290 F
291 F	292 F	293 F	294 F	295 F
296 T	297 F	298 F		

<Output of ALTLST.OUT>

ALTERNATIVE NUMBER LEGEND (P=1)

ALT NO	SITE NO
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12

ALTERNATIVE NUMBER LEGEND (P=2)

ALT NO	SITE NO
13	1- 2
14	1- 3
15	1- 4
16	1- 5
17	1- 6
18	1- 7
19	1- 8
20	1- 9
21	1-10
22	1-11
23	1-12
24	2- 3
25	2- 4
26	2- 5
27	2- 6
28	2- 7

29	2- 8
30	2- 9
31	2-10
32	2-11
33	2-12
34	3- 4
35	3- 5
36	3- 6
37	3- 7
38	3- 8
39	3- 9
40	3-10
41	3-11
42	3-12
43	4- 5
44	4- 6
45	4- 7
46	4- 8
47	4- 9
48	4-10
49	4-11
50	4-12
51	5- 6
52	5- 7
53	5- 8
54	5- 9
55	5-10
56	5-11
57	5-12
58	6- 7
59	6- 8
60	6- 9
61	6-10
62	6-11
63	6-12
64	7- 8
65	7- 9
66	7-10
67	7-11
68	7-12
69	8- 9
70	8-10
71	8-11
72	8-12
73	9-10
74	9-11
75	9-12
76	10-11
77	10-12
78	11-12

ALTERNATIVE NUMBER LEGEND (P=3)

ALT NO	SITE NO
79	1- 2- 3
80	1- 2- 4
81	1- 2- 5
82	1- 2- 6
83	1- 2- 7
84	1- 2- 8
85	1- 2- 9
86	1- 2-10
87	1- 2-11
88	1- 2-12
89	1- 3- 4
90	1- 3- 5
91	1- 3- 6
92	1- 3- 7
93	1- 3- 8
94	1- 3- 9
95	1- 3-10
96	1- 3-11
97	1- 3-12
98	1- 4- 5
99	1- 4- 6
100	1- 4- 7
101	1- 4- 8
102	1- 4- 9
103	1- 4-10
104	1- 4-11
105	1- 4-12
106	1- 5- 6
107	1- 5- 7
108	1- 5- 8
109	1- 5- 9
110	1- 5-10
111	1- 5-11
112	1- 5-12
113	1- 6- 7
114	1- 6- 8
115	1- 6- 9
116	1- 6-10
117	1- 6-11
118	1- 6-12
119	1- 7- 8
120	1- 7- 9
121	1- 7-10
122	1- 7-11
123	1- 7-12
124	1- 8- 9
125	1- 8-10
126	1- 8-11
127	1- 8-12

128	1- 9-10
129	1- 9-11
130	1- 9-12
131	1-10-11
132	1-10-12
133	1-11-12
134	2- 3- 4
135	2- 3- 5
136	2- 3- 6
137	2- 3- 7
138	2- 3- 8
139	2- 3- 9
140	2- 3-10
141	2- 3-11
142	2- 3-12
143	2- 4- 5
144	2- 4- 6
145	2- 4- 7
146	2- 4- 8
147	2- 4- 9
148	2- 4-10
149	2- 4-11
150	2- 4-12
151	2- 5- 6
152	2- 5- 7
153	2- 5- 8
154	2- 5- 9
155	2- 5-10
156	2- 5-11
157	2- 5-12
158	2- 6- 7
159	2- 6- 8
160	2- 6- 9
161	2- 6-10
162	2- 6-11
163	2- 6-12
164	2- 7- 8
165	2- 7- 9
166	2- 7-10
167	2- 7-11
168	2- 7-12
169	2- 8- 9
170	2- 8-10
171	2- 8-11
172	2- 8-12
173	2- 9-10
174	2- 9-11
175	2- 9-12
176	2-10-11
177	2-10-12
178	2-11-12
179	3- 4- 5
180	3- 4- 6

181	3- 4- 7
182	3- 4- 8
183	3- 4- 9
184	3- 4-10
185	3- 4-11
186	3- 4-12
187	3- 5- 6
188	3- 5- 7
189	3- 5- 8
190	3- 5- 9
191	3- 5-10
192	3- 5-11
193	3- 5-12
194	3- 6- 7
195	3- 6- 8
196	3- 6- 9
197	3- 6-10
198	3- 6-11
199	3- 6-12
200	3- 7- 8
201	3- 7- 9
202	3- 7-10
203	3- 7-11
204	3- 7-12
205	3- 8- 9
206	3- 8-10
207	3- 8-11
208	3- 8-12
209	3- 9-10
210	3- 9-11
211	3- 9-12
212	3-10-11
213	3-10-12
214	3-11-12
215	4- 5- 6
216	4- 5- 7
217	4- 5- 8
218	4- 5- 9
219	4- 5-10
220	4- 5-11
221	4- 5-12
222	4- 6- 7
223	4- 6- 8
224	4- 6- 9
225	4- 6-10
226	4- 6-11
227	4- 6-12
228	4- 7- 8
229	4- 7- 9
230	4- 7-10
231	4- 7-11
232	4- 7-12
233	4- 8- 9

234	4- 8-10
235	4- 8-11
236	4- 8-12
237	4- 9-10
238	4- 9-11
239	4- 9-12
240	4-10-11
241	4-10-12
242	4-11-12
243	5- 6- 7
244	5- 6- 8
245	5- 6- 9
246	5- 6-10
247	5- 6-11
248	5- 6-12
249	5- 7- 8
250	5- 7- 9
251	5- 7-10
252	5- 7-11
253	5- 7-12
254	5- 8- 9
255	5- 8-10
256	5- 8-11
257	5- 8-12
258	5- 9-10
259	5- 9-11
260	5- 9-12
261	5-10-11
262	5-10-12
263	5-11-12
264	6- 7- 8
265	6- 7- 9
266	6- 7-10
267	6- 7-11
268	6- 7-12
269	6- 8- 9
270	6- 8-10
271	6- 8-11
272	6- 8-12
273	6- 9-10
274	6- 9-11
275	6- 9-12
276	6-10-11
277	6-10-12
278	6-11-12
279	7- 8- 9
280	7- 8-10
281	7- 8-11
282	7- 8-12
283	7- 9-10
284	7- 9-11
285	7- 9-12
286	7-10-11

287	7-10-12
288	7-11-12
289	8- 9-10
290	8- 9-11
291	8- 9-12
292	8-10-11
293	8-10-12
294	8-11-12
295	9-10-11
296	9-10-12
297	9-11-12
298	10-11-12

BIBLIOGRAPHY

1. American Society of Photogrammetry. *Manual of Remote Sensing*. Falls Church, VA: American Society of Photogrammetry, 1975.

2. Arbel, Amy and Oren Shmuel S. "Generating Search Directions in Multiobjective Linear Programming Using the Analytic Hierarchy Process," *Socio-Economic Planning Sciences*: 369-373 (1986).

3. Bartezzaghi, E. et al. "A Search Tree Algorithm for Plant Location Problems," *European Journal for Operations Research*: vol. 7, 371-379 (1981).

4. Bate, Roger R. et al. *Fundamentals of Astrodynamics*. New York: Dover Publications Inc, 1971.

5. Berkeley, Capt Ron, Directorate of Air Requirements. Personal Interviews. National Defence Headquarters, Ottawa Canada, 19 March through 21 March 1990.

6. Berlin, G. N. et al. "Determining Ambulance-Hospital Locations for On-Scene and Hospital Services," *Environment and Planning*: vol.8, 553-561 (1976).

7. Brooke, Anthony et al. *GAMS: A User's Guide*. Redwood City, CA: The Scientific Press, 1988.

8. Chan, Yupo. Class handout distributed in OPER 663, Deterministic Methods in Operations Research. School of Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. March 1990.

9. ----- Class Lecture Notes for OPER 761, Multiple Criteria Decision Making. School of Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. September-December 1990.

10. ----- Class Lecture Notes for OPER 767, Network and Combinatorial Programming. School of Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. July-September 1990.

11. Chan, Yupo and Rowell, William. "Integrated Location and Routing Models - A Review of Model Formulations." Working Paper. Air Force Institute of Technology, Dayton, January 1990.

12. Church, R. and ReVelle C.S. "The Maximal Covering Location Problem," *Papers of the Regional Science Association*: vol. 32, 101-117 (1974).

13. Department of National Defence (Canada). *Challenge and Commitment: A Defence Policy for Canada*. Catalog No. D2-73/1987E. Ottawa: Canadian Government Publishing Centre, 1987.

14. Finley, Jim. Site Manager. Telephone Interview. GEODSS Site 1, Socorro, New Mexico. January 1991.

15. Gibson, Michael et al. "A Comparison of Interactive Multiple-Objective Decision Making Procedures," *Computers and Operations Research*: 97-105 (1987).

16. Hakimi, S. "Optimal Location of Switching Centers and the Absolute Center and the Medians of a Graph," *Operations Research*: vol.12, 450-459 (1964).

17. Headquarters Air Force Space Command, Director of Operations. *SATRAK*. Contract No. 0200121217, 0200121892 with Teledyne Brown Engineering, Colorado Springs, CO. Peterson AFB, CO.

18. Jeffrey, Lt Col G.I., Directorate of Air Requirements. Personal Correspondence. National Defence Headquarters, Ottawa Canada, 8 February 1990.

19. Jeffrey, Lt Col G.I., Directorate of Air Requirements. Personal Correspondence. National Defence Headquarters, Ottawa Canada, 9 May 1990.

20. Jensen, Paul A. *MICROSOLVE Network Flow Programming*. IBM-PC Student Version Software and User Manual. Oakland, California: Holden-Day Inc, 1985.

21. Jensen, Paul A. *MICROSOLVE Operations Research*. IBM-PC Student Version Software. Oakland, California: Holden-Day Inc, 1985.

22. Kelso, T.S. Assistant Professor of Space Operations. Personally Developed Satellite Pass Scheduling Software. School of Engineering, Air Force Institute of Technology, Wright Patterson AFB, OH, December 1990.

23. "LP83/MIP83 - A Professional Linear and Mixed Integer Programming System". Version Number 5.00. San Marino, California: Sunset Software, June 1985.

24. Lund, I.A. "Estimating the Probability of Clear Lines-of-Sight from Sunshine and Cloud Cover Observations". *Journal of Applied Meteorology, 4*: 714-722 (1965).

25. Malick, John D. et al. "Calibrated Analytical Modelling of Cloud-Free Intervals". *Proceedings of SPIE, 195*: 142-147 (1979).

26. Malo, Capt Frank, Chief Orbital Analyst. Telephone Interviews. Headquarters 1st Space Wing, Colorado Springs, Colorado. August 1989 through September 1990.

27. McCabe, J.T. *Estimating Mean Cloud and Climatological Probability of Cloud-Free-Line-of-Sight*. Technical Report 186, Air Weather Service, Scott AFB, Illinois. November 1965.

28. Michalowski, Wojtek and Szapiro Tomasz. "A Procedure for Worst Outcomes Displacement in Multiple Criteria Decision Making," *Computers and Operations Research*: 195-206 (1989).

29. Mirchandani, P.B. and Odoni, A.R. "Location of Medians on Stochastic Networks," *Transportation Science*: vol. 13, 85-97 (1979).

30. Mote, J. and Venkataramanan M. A. "A Comparative Multiobjective Programming Study," *Mathematical and Computer Modelling*: 719-729 (1988).

31. Nagarsenker B.N., Class Notes for MATH 596, Probability for Space Operations. School of Engineering, Air Force Institute of Technology, Wright Patterson AFB OH, July 1989.

32. ----- Personal Interview. School of Engineering, Air Force Institute of Technology, Wright Patterson AFB OH, January 1990.

33. Nemhauser G.L. and Wolsey L.A. *Integer and Combinatorial Optimization.* New York: John Wiley & Sons, 1988.

34. Odoni, A. "Stochastic Facility Location Problems," in *Stochastics in Combinatorial Optimization.* (G. Andreatta and F. Mason, editors). New Jersey: World Scientific, 1987.

35. Phillips, Don T. and Garcia-Diaz, Alberto. *Fundamentals of Network Analysis.* Englewood Cliffs, New Jersey: Prentice-Hall Inc, 1981.

36. Ringuest, Jeffrey L. and Rinks, Dan B. "Interactive solutions for the linear multiobjective transportation problem," *European Journal of Operational Research*: 96-106 (October 1987).

37. Roddy, Dennis. *Satellite Communications.* Englewood Cliffs, New Jersey: Prentice Hall, 1989.

38. Sorlin-Davis, Capt Janet, Assistant Chief Staff Meteorologist. Personal Interviews. Air Force Systems Command Aeronautical Systems Division, Wright-Patterson AFB OH, 5 April through 20 April 1990.

39. Steuer, Ralph E. *Multiple Criteria Optimization: Theory, Computation, and Application.* New York: John Wiley & sons, 1986.

40. Swain, R.W. "A Parametric Decomposition Approach for the Solution of Uncapacitated Location Problems," *Management Science*: vol. 21, no. 2, 189-198 (1974).

41. *The Canadian Encyclopedia.* Edmonton: Hartig Publishers, 1985.

42. Thompson, Roy C., Chief of Optical Surveillance Systems. Telephone Interview. Headquarters Air Force Space Command, Colorado Springs CO, 15 August 1989.

43. Toregas, C. *Location Under Maximal Travel Time Constraints.* Ph.D dissertation. Cornell University. 1971.

44. Toregas, C. and ReVelle C. "Optimal Location Under Time or Distance Constraints," *Regional Science Association Papers*: vol. 28, 133-143 (1972).

45. Trewartha et al. *Elements of Geography.* New York: McGraw Hill Book Company, 1967.

46. United States Space Command. *Space Surveillance Network Tactical Requirements.* UR 55-12. Colorado Springs: HQ USSPACECOM, 1 Nov 88.

47. USAF Environmental Technical Applications Center (ETAC/DNY). Telephone Interviews. Scott AFB, Illinois 62225-5438, September through December 1990.

48. USAF Environmental Technical Applications Center (ETAC/DNY). Personal Correspondence. Scott AFB, Illinois 62225-5438, 17 December 1990.

49. Vanderpooten, Daniel. "The Interactive Approach in MCDA: A Technical Framework and Some Basic Conceptions," *Mathematics and Computer Modelling*: 1213-1220 (1989).

50. Yu, Po-Lung. *Multiple-Criteria Decision Making.* New York: Plenum Press, 1985.

VITA

Major Pierre J. Forgues was born on 23 March 1955 in Ottawa, Canada. He graduated from the University of Toronto in 1978 with a Bachelor of Science degree in Survey Science. In 1980, he joined the Canadian Forces as an Air Weapons Controller in North Bay, Ontario. A year later, he was posted to the Arctic where he completed a one-year tour serving on the DEW line. He then returned to North Bay to instruct on the Basic Air Weapons Controller Course for two years before being posted to the Aerospace Systems Course in Winnipeg, Manitoba. Following this one-year aerospace engineering course, he moved to Cold Lake, Alberta in 1985 to serve as Range Operations Officer, and, a year later, as Officer Commanding of the Primrose Lake Evaluation Aircraft Test Range. In 1987, he was promoted to the rank of Major and was posted to United States Space Command in Colorado Springs to work in space surveillance operations. He left Colorado Springs in May 1989 to attend the Air Force Institute of Technology Graduate Space Operations Program.

Major Forgues received a Master of Arts degree in Education Administration from Central Michigan University in 1986. He is a member of the Tau Beta Pi national engineering honor society.